ENCICLOPEDIA PRACTICA DE LA

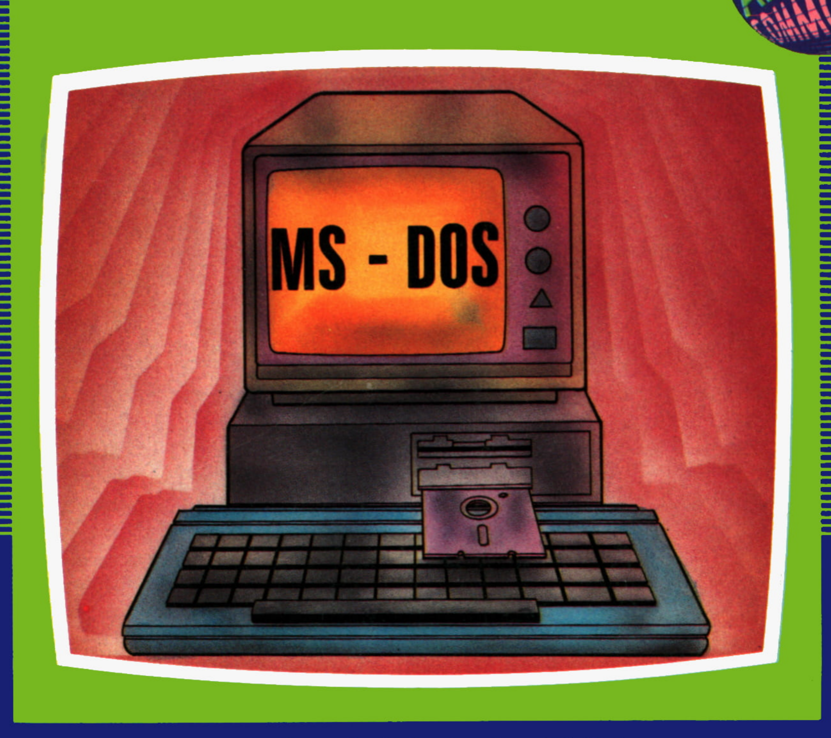
INFORMATICA APLICADA

36

MS-DOS

AIA

SINCLAIR AMSTRAD COMMODORE IBM



EDICIONES SIGLO CULTURAL

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

36

MS-DOS

EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S.A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Directores de la colección:
MANUEL ALFONSECA, Doctor Ingeniero de Telecomunicación y Licenciado en Informática
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño y maquetación:
BRAVO-LOFISH.

Dibujos:
JOSE OCHOA Y ANTONIO PERERA.

Tomo XXXVI. **MS-DOS**
AULA DE INFORMATICA APLICADA
SOLEDAD TAMARIZ-MARTELL, diplomada de Telecomunicación

Ediciones Siglo Cultural, S.A.

Dirección, redacción y administración:
Pedro Teixeira, 8, 2.ª planta (Iber a Mart-I). Teléf.: 810 52 13. 28020 Madrid.

Publicidad:
Gofar Publicidad, S.A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:
COEDIS, S.A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América. 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

ISBN del tomo: 84-7688-142-8
ISBN de la obra: 84-7688-018-9.

Fotocomposición:
ARTECOMP, S.A. Albarracín, 50. 28037 Madrid.

Imprime:
MATEU CROMO. Pinto (Madrid).

Depósito legal: M. 20.037-1987
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S.A.
Pedro Teixeira, 8, 2.ª planta (Iberia Mart-I). Teléf.: 810 52 13. 28020 Madrid.

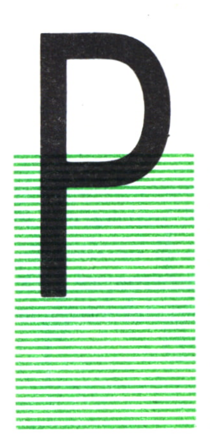
Julio, 1987
P.V.P. Canarias: 365,-

INDICE

Los programas que aparecen en este libro funcionan en los ordenadores:

IBM-PC, XT, AT y compatibles.
AMSTRAD-464, 664, 6128, 1512.
SINCLAIR-SPECTRUM 48 K, 128 K, PLUS, PLUS 2.
MSX-Todos los modelos.
COMMODORE-CBM 64 y CBM 128.

INTRODUCCION 1



POSIBLEMENTE al recibir con su ordenador personal una copia del MS-DOS surgió la cuestión de su utilidad, pronto el hecho de descubrir un sistema operativo en esas curiosas siglas le abrió una gran cantidad de posibilidades que en principio desconocía.

El objetivo de este libro va a ser introducirle en el mundo del DOS, o si lo prefiere ayudarle a fijar los conocimientos que ya posee y ofrecerle ideas y nuevas posibilidades.



¿QUE ES EL MS-DOS?

En primer lugar vamos a aclarar el significado de las siglas, ya que esto puede ayudar enormemente a comenzar un conocimiento del programa. DOS es el acrónimo de Disk Operating System (sistema operativo de disco). Un sistema operativo es un conjunto de programas que dirigen y ayudan en el control de todas las partes del ordenador, los conocimientos que deben tenerse de un sistema operativo dependen de las necesidades de cada usuario, sin embargo, es imprescindible una base que nos permita evitar errores en el manejo de nuestro ordenador o, lo que también es grave, la pérdida de posibilidades del mismo por desconocimiento de ellas.

La gran cantidad de funciones que permite el DOS, sugieren que su aprendizaje se realice mediante la práctica: no pierda nunca de vista su guía de usuario, en la que puede encontrar en cualquier momento la forma de utilización de un comando. Su consulta no indica falta de conocimientos, más bien un perfecto sistema de trabajo.

Otro punto a señalar son las diferentes versiones existentes del DOS; como veremos, esto se debe a las revisiones que se han realizado del programa para añadirle posibilidades y para obtener un mayor rendimiento

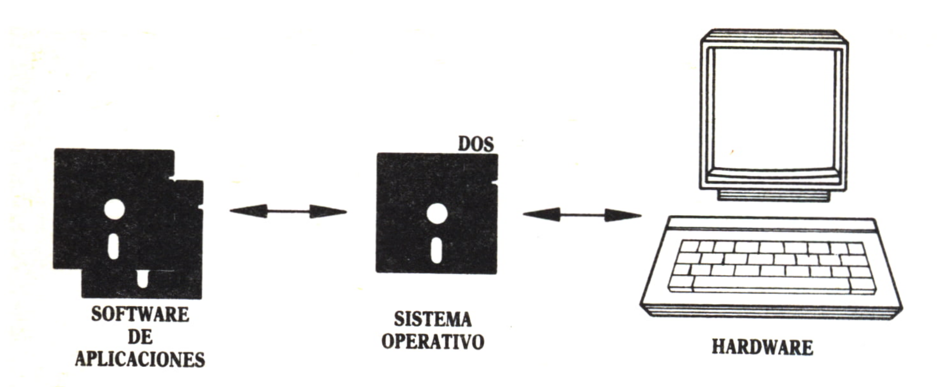


Fig. 1. El sistema operativo es un conjunto de programas que controla cada una de las partes del ordenador. Sin él no funcionarán los programas de aplicaciones.

del hardware, tendremos, de alguna forma, una pescadilla mordiéndose la cola: el progreso del hardware requiere un progreso en el software y al progresar el software necesitamos una mayor capacidad en el hardware.

Cada vez que encendemos el ordenador, al iniciar el sistema, el DOS nos indica con qué versión estamos trabajando, en principio debemos saber que todas las versiones son compatibles.



INICIOS DEL MS-DOS

El origen del DOS puede encontrarse en un largo proceso de progreso informático que vamos a resumir desde 1972 en los siguientes puntos:

— Aparición del microprocesador.
— Comienzos del software para microprocesador: Lenguaje máquina.
— Diseño de ensambladores, que permiten el uso de mnemónicos.
— Aparición de los debuggers, que permiten encontrar errores en los programas.

Después de estos primeros pasos se consideró necesario hallar el modo adecuado de introducir los programas en el microprocesador. El método utilizado hasta el momento consistía en una lectora de tarjetas y una perforadora para traducir los resultados. Evidentemente, la integración del hardware se veía deteriorada por la imposibilidad de realizar una perforadora de reducidas dimensiones; sin embargo, los intentos por conseguir una lectora/perforadora de bolsillo orientaron a los ingenieros en un nuevo campo:

— Aparece la unidad de disco y el floppy o diskette. Este podía almacenar 246.784 bytes de información, el equivalente a una milla de tarjetas de papel.

Evidentemente, era necesario crear un programa que rigiera el comportamiento interno de este sistema de almacenamiento de información, su interacción con el ordenador y la del ordenador con el resto de los periféricos:

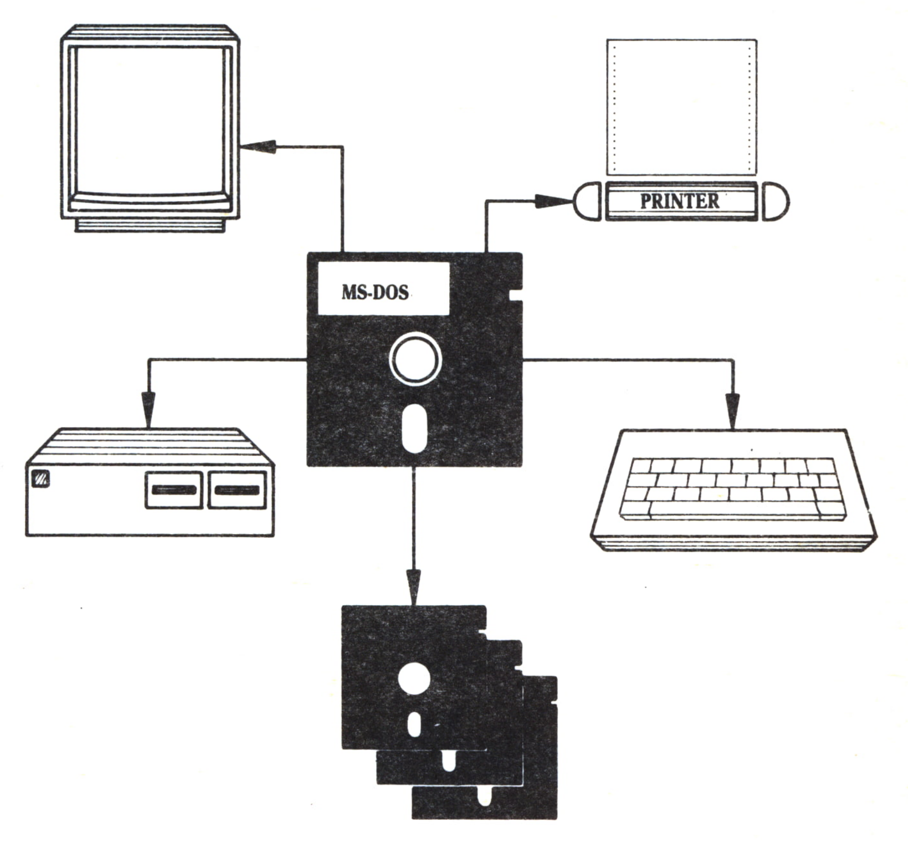


Fig. 2.

— Se creó un nuevo sistema operativo al que se llamó sistema operativo de disco. El nombre era debido, principalmente, a la importancia de control de las unidades de disco en este programa.

En torno a este principio se lanza, en 1975, el primer microcomputador con diskettes como unidad de almacenamiento, cuyo control estaba reservado a un programa denominado CP/M (Control Program for Microcomputers). A partir de aquel año este sistema operativo de disco fue creciendo de forma estándar para microprocesadores de Intel (CPU 8080 y 8085) y de Zilog Corporation (CPU Z80).

Con la aparición del 8086, cuya palabra es de 16 bits y el 8088 que permite palabras de 8 y 16 bits, el CP/M perdió su campo de estandarización

y se vio la necesidad de un nuevo sistema operativo de disco que resolviera la utilización correcta de los nuevos microprocesadores. El autor fue Tim Paterson, y el nombre del programa, 86-DOS.

Poco después, IBM lanzó al mercado su ordenador personal al que quiso dotar de un potente y estándar lenguaje BASIC, así como el sistema operativo que acababa de aparecer. El 86-DOS y el BASIC-86 eran comercializados por Microsoft. Con este lanzamiento el MS-DOS o PC-DOS pronto llegó a ser el sistema operativo de disco más popular.

Las diferentes versiones que fueron apareciendo del MS-DOS iban perfeccionando el sistema e introduciendo elementos adicionales muy influenciados por el sistema operativo XENIX (una versión del UNIX).

La aparición del PC-XT y del disco duro (hard disk) presentaba una nueva orientación para el sistema operativo que fue resuelta con el sistema de directorios jerárquicos.



ARRANCANDO EL SISTEMA

Vamos a hacer una pequeña diferenciación para ver el modo de arrancar el sistema en los ordenadores con disco duro y con una (o dos) unidades de disco.

1. *Disco duro:* Si su disco está instalado, la forma de arrancar el sistema consiste en encender el ordenador manteniendo las unidades adicionales de disco que posea abiertas.

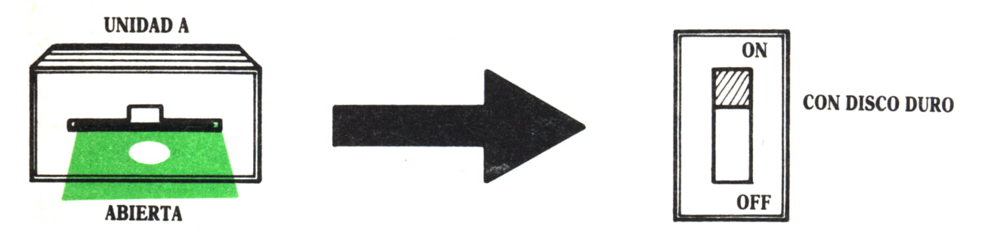


Fig. 3.

2. *Unidades de disco:* Introduzca el disco del MS-DOS en la unidad A y proceda a conectar el ordenador.

Lo primero que veremos, tras un sonido del ordenador, será el siguiente mensaje:

La fecha actual es Mar 1-01-1980
Introduzca nueva fecha (dd-mm-aa):

El guión intermitente que aparece tras los dos puntos es el cursor que nos muestra dónde se va a situar el texto que se teclee y la situación de

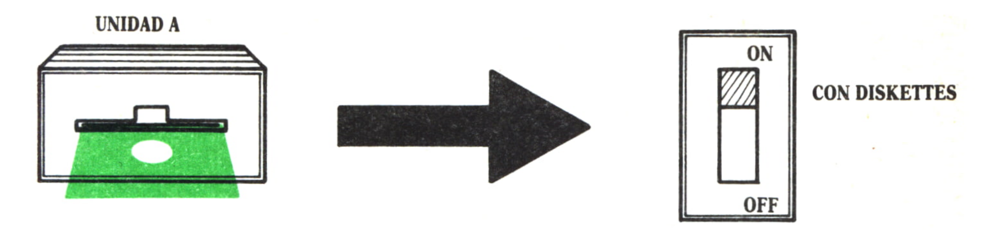


Fig. 4.

espera del ordenador. Introduzca la fecha actual correctamente y separada por guiones, por ejemplo:

1 de abril de 87

Pulse [Intro] para que acepte la nueva fecha. El proceso continúa de la siguiente forma:

La fecha actual es Mar 1-01-1980
Introduzca nueva fecha (dd-mm-aa):1-04-87
La hora actual es 0:01:15.83
Introduzca nueva hora:

Del mismo modo que antes el sistema está a la espera de la nueva hora, introducimos la hora en la que estamos 20:45.

Después de introducir la fecha y la hora, el DOS saca un mensaje en el que pone el nombre del programa y la versión con la que estamos trabajando. Aparece delante del cursor la letra A y el símbolo >, indicando que el sistema está a la espera de nuevas instrucciones. (Si trabajamos con disco duro aparecerá C>.)

La fecha actual es Mar 1-01-1980
Introduzca nueva fecha (dd-mm-aa):1-04-87
La hora actual es 0:01:15.83
Introduzca nueva hora: 20:45

IBM Personal Computer - Versión DOS 3.10

A>

Si su ordenador dispone de reloj interno, este proceso va a variar en algunos momentos, ya que no será necesario introducir la nueva fecha y hora si las que nos pone el ordenador son correctas.

Si por una falta de alimentación prolongada el reloj no da la hora correcta o la fecha se ha alterado, inicialice el reloj con el comando correspondiente.

```
Phoenix ROM BIOS Ver 2.27
Copyright (c) 1984,1985.1986 Phoenix Technologies Ltd
All Rights Reserved

YANGTECH.INC


A>keybsp

A>reloj/s

**  Wanper Reloj V 1.3  **

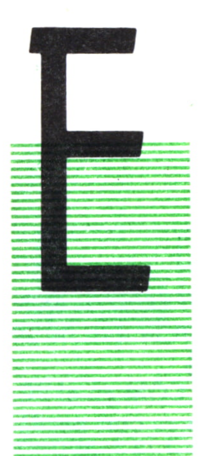
La Hora es       10:29:14.76

La fecha es      Mie 04-19-00

A>
A>
```

Fig. 5.

ESTRUCTURA DEL DOS 2



INTRODUCCION



EL MS-DOS está compuesto principalmente por tres partes:

— El sistema de entrada/salida.
— El procesador de comandos.
— Las utilidades.

El sistema de entrada/salida nos va a permitir escribir caracteres (o cualquier tipo de dato), visualizarlos en el monitor, imprimirlos en la impresora, almacenarlos en el disco y cargarlos en el mismo.

El procesador de comandos contiene una gran variedad de posibilidades que actúan como programas independientes y nos van a permitir copiar, visualizar o ejecutar los ficheros.

Por último, las utilidades también forman un conjunto de programas independientes que nos facilitarán darle formato a un disco, ver el espacio libre que posee, comparar archivos, etc.



EL BIOS

BIOS es el acrónimo de Basic Input Output System (sistema básico de entrada salida) y constituye el software que realiza la función de entrada/salida, es decir, las funciones que relacionan el ordenador con sus periféricos.

Cada periférico posee una tarea y un modo de comunicarse con el ordenador diferente, por ello, antes de la comunicación, el ordenador debe prepararse para controlar el periférico.

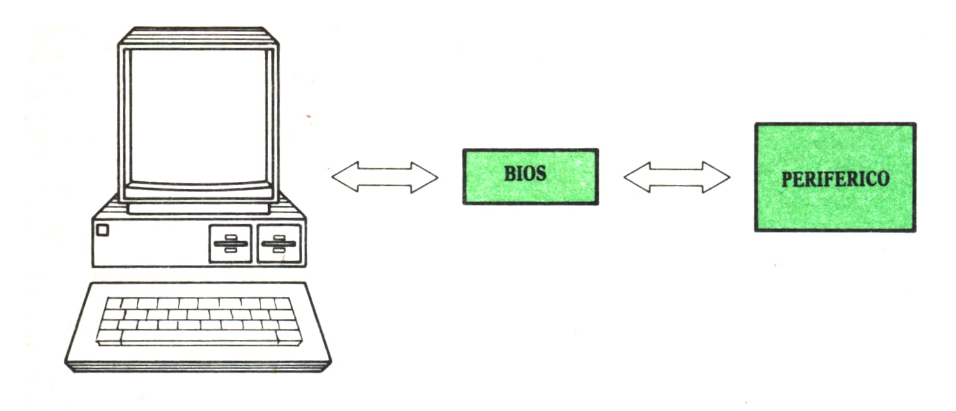


Fig. 1.



FICHERO IBMBIO.COM

El BIOS del PC se encuentra localizado en dos partes:

— La mayor parte de las rutinas de comunicación se encuentran localizadas en una memoria ROM (memoria sólo de lectura, en la que no se puede escribir) y reciben el nombre de ROM BIOS o RIOS.

— El archivo IO.SYS, que recibe el nombre de IBMBIO.COM en el PC, en él se almacenan las rutinas de control de comunicaciones con periféricos. Este archivo se encuentra en el disco del DOS.



FICHERO IBMDOS.COM

Otra parte importante del sistema de entrada/salida es el MSDOS.SYS, que en el caso del PC recibe el nombre de IBMDOS.COM. En este archivo se encuentra la mayor parte del sistema operativo y las rutinas para control de la información que se transfiere entre los periféricos y el ordenador. Pueden distinguirse dos sistemas de archivo diferentes: uno para las unidades de disco y otro para el resto de los dispositivos.

Pasamos a ver el método que utiliza el ordenador en cada uno de estos sistemas:

La impresora, el monitor, el teclado, el teletipo y los modems son periféricos que trabajan enviando o recibiendo un carácter por unidad de tiempo. Sin embargo, las unidades de disco trabajan con bloques de caracteres.

Por ello, los sistemas utilizados son diferentes según estos dos grupos de dispositivos y el DOS debe tener en cuenta a cuál nos estamos refiriendo para acudir a un grupo de programas de entrada /salida o a otro.

En el primer grupo (un carácter por unidad de tiempo) el DOS debe

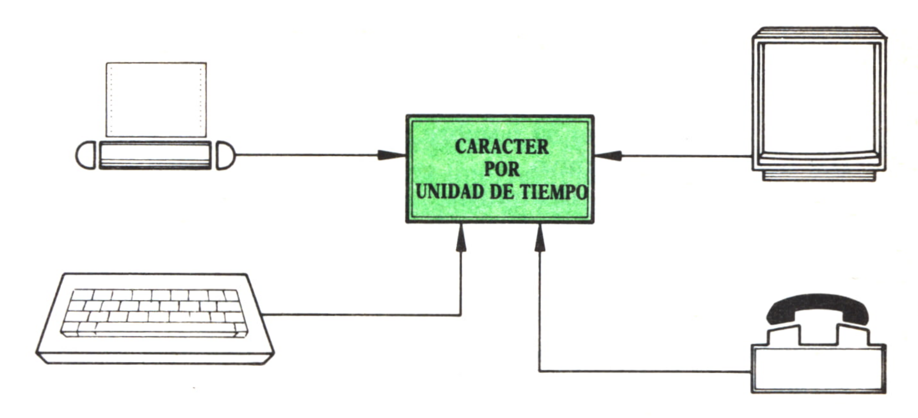


Fig. 2.



Fig. 3.

chequear el dispositivo para ver cuándo quiere enviar un carácter o, en caso de ser un dispositivo de salida, para ver si está conectado y si el carácter puede ser enviado. Una vez confirmado este punto el carácter puede pasar a la línea correspondiente.

Un resumen de las funciones que el DOS debe realizar en este grupo de periféricos es:

— Leer un carácter de un dispositivo.
— Enviar un carácter a un dispositivo.
— Chequear si el dispositivo sobre el que se quiere leer o escribir está activo (ready).
— Recoger una serie de caracteres del teclado.
— Enviar una serie de caracteres al monitor.
— Recoger, enviar o actualizar la fecha y la hora.

En el segundo grupo (dispositivos orientados a bloques) el DOS debe situar la cabeza de lectura o escritura en el lugar correspondiente del disco y extraer, o añadir, un bloque de información en dicha posición.

Un resumen de las funciones que el DOS debe realizar en este grupo de periféricos es:

— Crear un archivo.
— Abrir un archivo ya existente para leer o escribir en él.

— Escribir información en un archivo.
— Leer información desde un archivo.
— Cerrar un archivo.
— Buscar un archivo por su nombre.
— Dar nombre a un archivo.
— Dar la dimensión de un archivo.
— Cambiar el nombre o atributos de un archivo.
— Trasladar o borrar un archivo.
— Inicializar el sistema.
— Indicar la unidad de disco que se está utilizando.
— Especificar el espacio libre en disco.
— Cargar y ejecutar programas.
— Crear un directorio jerárquico en el sistema.
— Establecer divisiones de memoria para le ejecución de programas.
— Control libre de memoria.

Esta lista puede darle una visión general de las diferentes tareas que puede desempeñar el DOS en entrada/salida y que permiten que el usuario se desentienda del control de las mismas. Posiblemente, algunas de ellas no estén muy claras por el momento, pero las iremos comentando a lo largo del libro.



EL PROCESADOR DE COMANDOS

Ya se mencionó en la introducción de este capítulo que constituye una de las partes fundamentales del DOS. Recibe el nombre de COMMAND.COM y es el programa mediante el que nos vamos a comunicar con el DOS para indicarle las tareas que queremos que realice.

COMMAND.COM es el encargado de escribir el «PROMPT» en la pantalla (ejemplo: A>), este es un indicador de la unidad de disco en la que se está trabajando y constituye la toma de la palabra por el usuario en el diálogo con COMMAND.COM, es decir, cuando aparece el prompt podemos utilizar el ordenador para escribir el comando que deseamos, una vez pulsada la tecla [Intro] el ordenador pasa a la ejecución de dicho comando, y el prompt desaparece hasta que el procesador finaliza su tarea y nos vuelve a conceder la palabra. La correcta traducción de prompt es incitador o generador de respuesta, se conoce también como carácter de interacción, sin embargo, como tantas otras palabras del campo de la informática, se prefiere mantener la palabra inglesa.

COMMAND.COM está formado por varias partes, entre ellas vamos a destacar el control de interrupciones y el control de errores.

Control de interrupciones: Cuando un dispositivo periférico requiere atención, por ejemplo, para enviar caracteres, envía lo que se llama una

señal de petición de interrupción, COMMAND.COM se encarga de interrumpir la tarea en curso, almacenar los parámetros necesarios para poder reanudarla más tarde y dar paso a la petición del periférico. Una vez terminada la transferencia correspondiente se encarga de continuar la acción en el mismo punto en que la interrumpió.

Control de errores: Principalmente se centra en los errores del disco, divisiones por cero o errores de sintaxis. COMMAND.COM visualizará un mensaje de error que permite detectar el motivo del mismo.

Vamos a destacar tres características importantes de este programa:

1. Al terminar la ejecución de un programa en memoria, COMMAND.COM se chequea a sí mismo recargándose si es necesario. Para ello si no trabajamos con disco duro y el disco del DOS no está en la unidad A nos pedirá:

Insertar disco con COMMAND.COM en unidad A

2. Existen dos porciones de COMMAND.COM. Una de ellas actúa como un programa residente en memoria, es decir, se encuentra siempre en la RAM del ordenador. La otra puede ser cargada sólo cuando se va a utilizar de forma que no esté ocupando un espacio innecesario en memoria. Esta es la parte que COMMAND.COM se encarga de recargar cuando su espacio ha sido utilizado por otro programa, como se indicaba en el apartado anterior.

En la parte residente se encuentran los comandos más típicamente utilizados, como, por ejemplo:

COPY permite copiar archivos del mismo disco y entre discos.
DATE actualiza o muestra la hora del sistema.
DEL borra archivos de un disco.
RENAME cambia el nombre de un fichero de un disco.
TIME actualiza o muestra la hora del sistema.
TYPE visualiza en la pantalla el contenido de un fichero.

Debemos añadir a esta lista los subcomandos o ficheros batch ECHO, IF, FOR..IN..DO, GOTO, REM y PAUSE.

Y los comandos de directorios jerárquicos CHDIR, MKDIR y RMDIR.

Más adelante veremos con detalle los comandos mencionados, de momento lo importante es que todos ellos se encuentran residentes en memoria y no es necesario cargar ningún programa para ejecutarlos.

3. COMMAND.COM tiene la misión de que el DOS cargue los programas indicados por el usuario en memoria y pasa el control sobre los mismos directamente al ordenador hasta que la ejecución haya terminado.

Un ejemplo del proceso es el siguiente: Supongamos que desea utilizar un procesador de textos que se llama PTXT. Se teclea PTXT junto al prompt que indica en qué unidad de disco debe estar el diskette con el pro-

grama y se pulsa [Intro]. En ese momento, COMMAND.COM inicia su búsqueda entre sus comandos, como PTXT no es uno de sus comandos va a la unidad de disco activa y busca un programa llamado PTXT.COM o PTXT.EXE. Si encuentra el programa, lo carga en memoria, da control al ordenador y pasa a ejecutarlo.

Nos hemos comunicado con el COMMAND.COM, que, a su vez, se ha comunicado con el BIOS, y éste se ha encargado de la comunicación directa con los periféricos (en este caso con la unidad de disco).



ARRANQUE DEL DOS

Sabemos ya, por el capítulo anterior, cómo debemos arrancar nuestro ordenador con el DOS, pero veamos ahora cómo se carga el programa y los pasos que sigue para quedar instalado en la memoria del ordenador.

En el momento de conectar el sistema, o realizar un reset del mismo (se consigue pulsando simultáneamente las teclas [Control],[Alt] y [Del].), el ordenador va a ejecutar un pequeño programa situado en la memoria ROM y que se conoce con el nombre de BOOTSTRAP. Es el primer programa y da paso a la lectura de la primera parte existente en el diskette de la unidad A (La C si disponemos de disco duro), parte que se sitúa en la RAM.

El programa BOOTSTRAP busca en la lista de programas del disco el IBMBIO.COM y el IBMDOS.COM. Si estos programas coinciden con el primero y segundo de la lista el proceso continúa tras la carga en memoria de los mismos, en caso contrario se produce un fallo que dará lugar a un error.

El siguiente paso se denomina inicialización, se produce de la siguiente forma: el programa BOOTSTRAP cede el control al IBMBIO.COM, éste se encarga de chequear los dispositivos que están conectados al ordenador, si obtiene un resultado positivo inicializa el estado del periférico.

Para comprobar este proceso puede conectar su ordenador estando la impresora encendida, observará que se escucha un sonido en la impresora y el carro se coloca en la posición inicial.

Cuando todos los periféricos han sido inicializados, IBMBIO.COM busca en el directorio principal del disco el fichero CONFIG.SYS, si lo encuentra lo carga en memoria y procede a su ejecución. Este fichero contiene la configuración del sistema (unidades de disco que posee, cúal es la principal, si posee disco duro, si tiene conectada alguna impresora, si posee tarjeta de gráficos...).

Algunos programas de aplicación a dispositivos accesorios requieren la inclusión de ciertos comandos en este archivo, pero no se preocupe por ello, cuando esto sea necesario vendrá correctamente explicado en el manual correspondiente.

Llegado este punto podemos considerar finalizada la tarea de carga del IBMBIO.COM, éste permite una relocalización en la zona correcta de memoria del IBMDOS.COM. Este realiza su cometido permaneciendo residente en memoria (al menos una parte, como ya hemos visto) y destina una zona de memoria para cargar el COMMAND.COM, que pasa a tener control del programa. Cualquier comando que se introduzca a partir de este momento podrá ser interpretado por el COMMAND.COM.

En este momento, el DOS está completamente cargado y puede considerarse terminado su proceso de inicialización.

Antes de dar el control al usuario el DOS busca el fichero AUTOEXEC.BAT en el diskette. Si lo encuentra, procede a la ejecución de los comandos que lo forman y que, al poder ser definidos por el usuario, permiten establecer la inicialización del sistema de una forma concreta. Un ejemplo de ello sería un AUTOEXEC.BAT que definiera un teclado español, fecha y hora, y el nombre de un procesador de textos que se encuentre en el disco, con ello podremos conseguir una ejecución directa del mismo cada vez que encendamos el ordenador debido a que el DOS se encarga de lo necesario.

Si su ordenador posee un disco duro el proceso es muy similar, el programa BOOTSTRAP chequea la unidad A, al encontrar que está abierta pasa la búsqueda a la unidad C, que es la del disco duro.

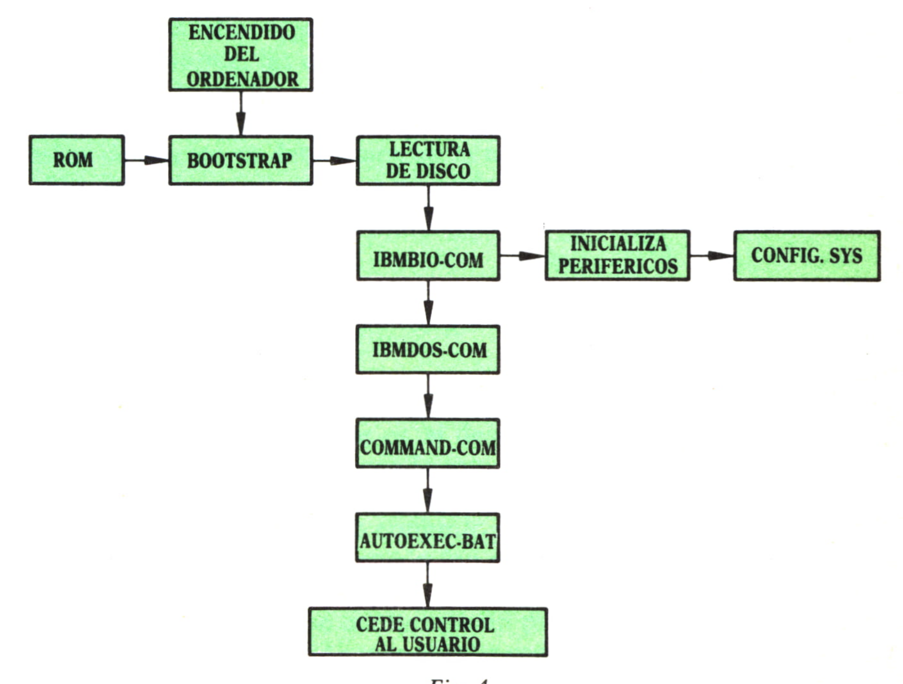
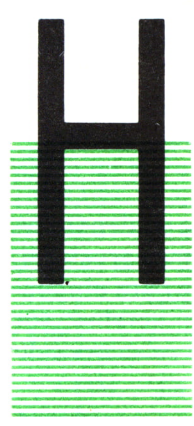


Fig. 4.

PRIMEROS PASOS CON EL DOS 3



INTRODUCCION



ASTA ahora hemos visto los principios de funcionamiento del DOS, posiblemente vayan quedando más claros a medida que con el uso del DOS pueda ir sacándoles partido y vea su funcionamiento en la práctica. También es probable, por otra parte, que se encuentre algo desilusionado: nos encontramos en el capítulo 3, conocemos los principios básicos del programa en cuestión, pero lo único que sabemos hacer hasta ahora es encender el ordenador, algo que, por supuesto, ya se sabía de antemano.

No se desanime, las cosas no son tan negras, los conocimientos que le hemos ofrecido en los capítulos anteriores son fundamentales para comprender el porqué de lo que viene en los siguientes. Ante todo somos conscientes de que no se debe empezar la casa por el tejado.

Vamos a pasar, por tanto, a aprender a utilizar el DOS. Veremos el manejo de ficheros, discos y periféricos, así como los comandos más utilizados y necesarios.



LOS ARCHIVOS

Llegamos a una palabra que hemos utilizado ya en varias ocasiones sin dar una definición de lo que es, si desconocía el significado informático de esta palabra, no habrá quedado muy lejos de comprenderla a la perfección. Pero demos una definición que le permita fijar la idea.

El archivo o fichero es un conjunto de información con un cierto nombre que se encuentra almacenado (en el caso del DOS se almacena en los discos).

En general, un archivo contiene o un programa o datos, pero podemos establecer una clasificación más amplia según sean archivos de texto (contienen información simbólica que se puede leer), archivos de comandos

(contienen instrucciones necesarias que el DOS precisa para llevar a cabo los comandos, serán tratados en su ejecución por COMMAND.COM como ya hemos visto) y archivos de programas de aplicación (como un procesador de texto, una hoja electrónica, un paquete integrado) cuyos datos de trabajo se almacenan en un archivo de datos de aplicación.



COMO SE NOMBRAN LOS ARCHIVOS

Un archivo puede tener un nombre formado por hasta ocho caracteres, éstos podrán ser letras o números o algún carácter especial como:

! @ # $ % & ( ) - _ {}‘’

No se pueden utilizar los siguientes caracteres:

ˆ = + / [ ] ” : ; , . ? *

Al nombre se le puede añadir un punto y tres letras que se denomina extensión y que nos facilitará el contenido del fichero con más precisión. Los caracteres válidos para la extensión son los mismos que para el nombre. La extensión distingue un archivo de otro al igual que el nombre, por tanto, CARTA1 y CARTA1.DTA serán distintos archivos, del mismo modo que serán distintos CARTA1.DTA Y CARTA1.TXT.

Veamos algunos ejemplos aclaratorios:



Nombres inválidos

1987PRESUPUESTO	El nombre es demasiado largo.
.DTA	No tiene nombre de archivo.
PRESUPTO.1987	La extensión es demasiado larga.
CARTA JT.ENE	No están permitidos los huecos.
CARTA,JT.ENE	No está permitida la coma.
CARTA.JT.ENE	El punto sólo es válido para separar la extensión.



Nombres válidos

1987PRES	P
CARTA.DTA	XXXXXXXX.XXX
PRESUPTO.87	123XXX.MES
CARTAJT.ENE	MSDOS1.LIB
CART{JT}.ENE	MSDOS2.LIB

Los nombres cortos como P resultan más fáciles y rápidos de escribir, sin embargo, debemos intentar poner nombres y extensiones tan descrip-

tivas como nos sea posible, de forma que nada más verlo sepamos relacionarlo con su contenido:

CARTAJT.ENE es el fichero que contiene la carta escrita a José Tórtolo en el mes de enero.

PRESUPTO.87 es el fichero que contiene el presupuesto del 87.

MSDOS1.LIB es el fichero que contiene el primer capítulo (1) del libro (.LIB) de MS-DOS.

Existen extensiones creadas por el DOS o creadas para que el DOS sepa el tipo de información que posee. Se debe evitar dar extensiones con estos nombres a nuestros archivos. Las extensiones más comúnmente utilizadas son:

ASM — Se trata de un fichero fuente de ensamblador.

BAK — Abreviatura da backup. Contiene la penúltima versión de un archivo de texto.

BAS — Archivo de programa escrito en BASIC.

BAT — Abreviatura de batch. Contiene un fichero de texto con comandos del DOS que se ejecutan al escribir el nombre del archivo.

BIN — Contiene un programa en código binario.

C — Contiene un fichero fuente de C.

COB — Contiene un programa en COBOL.

COM — Es la abreviatura de COMMAND. Se refiere a un archivo de comandos.

DAT — Contiene un fichero de datos.

DOC — Contiene un fichero documento (texto).

DTA — Contiene un fichero de datos.

EXE — Contiene un programa ejecutable.

FOR — Contiene un fichero en FORTRAN.

HLP — Contiene un fichero de ayuda.

LIB — Contiene un programa de librería.

MSG — Archivo de mensajes.

PAS — Contiene programa en PASCAL.

SYS — Identifica un archivo de uso exclusivo del DOS. Corresponde a la configuración del sistema.

TMP — Se trata de un archivo temporal.

TXT — Archivo de texto.

$$$ — Archivo temporal o incorrectamente almacenado.

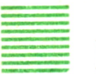
Algunas de estas extensiones pueden considerarse de carácter tan estándar que son utilizadas por la inmensa mayoría de los programas de aplicaciones. No obstante, las extensiones que utiliza un programa para trabajar vienen especificadas en su manual, y el usuario no debe utilizarlas para evitar errores.



Los discos

Los discos son los periféricos más importantes de su ordenador, el teclado y el monitor constituyen las unidades de entrada y salida, y la impresora y el modem son elementos adicionales para extender la capacidad del sistema. Sin embargo, los discos permiten trabajar con gran cantidad de información que puede ser almacenada y cargada y, algo importante, constituyen el eje central de actuación del DOS.

Vamos a establecer una diferencia entre un floppy o diskette y un disco duro.



DISKETTES

Vamos a considerar los diskettes como archivadores en los que almacenaremos nuestros ficheros. Su manejo va a necesitar una serie de cuidados.

Veamos, en primer lugar, la utilización de un diskette. Se trata de materiales bastante duraderos, dependiendo su vida útil de lo poco o mucho que se empleen y del modo en que se traten. Mantenga siempre las siguientes precauciones:

— Evite tocar la superficie del floppy que se ve a través del orificio de lectura, manténgalos siempre que no los utilice en la funda protectora para evitar que el polvo pueda perjudicar la lectura.

— Guarde los diskettes lejos de cualquier imán o fuentes de campo magnético, como puede ser el teléfono. Cuidado también con las calefacciones, no los deje muy cerca del radiador.

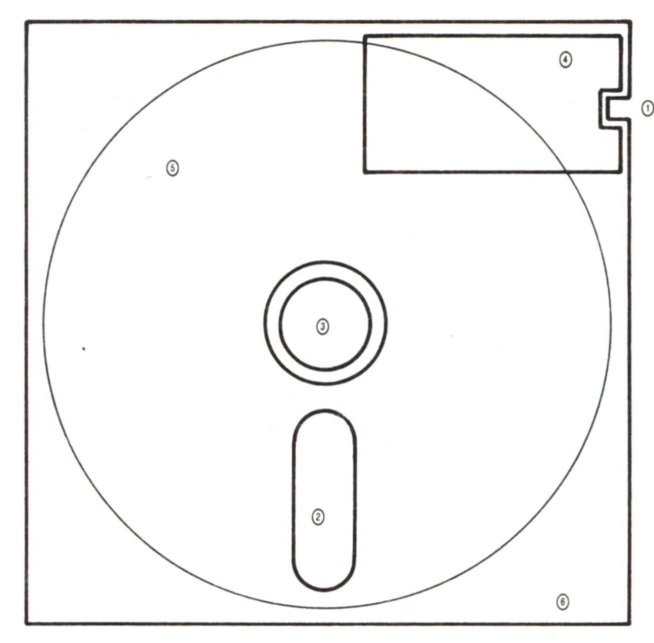
— No los doble ni coloque objetos sobre ellos, aunque tengan forma de posavasos ¡no es recomendable utilizarlos como tales!

— No escriba los nombres en las etiquetas con lápiz o bolígrafo, utilice un rotulador y escriba sin presionar.

Otro punto muy aconsejable son las copias de seguridad, evitan el riesgo de perder un volumen de información si el diskette se deteriora por cualquier motivo.

En segundo lugar vamos a ver cómo se almacena la información en el diskette:

Lo que denominamos diskette o floppy consta de dos partes: un disco delgado de plástico revestido de un material magnético y una protección de plástico. En él observamos una abertura central para el eje del motor que lo hace girar, una abertura sobre la que lee y escribe la cabeza, una muesca pequeña que nos permitirá, al cubrirla, proteger el diskette contra la escritura.



① Muesca para evitar el borrado.
② Abertura para lectura/escritura.
③ Abertura para el eje de giro.
④ Etiqueta.
⑤ Disco interno.
⑥ Cubierta protectora externa.

Fig. 1.

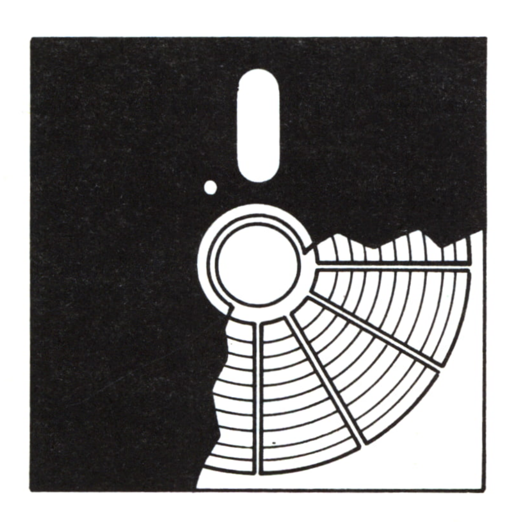


Fig. 2.

Conviene que en sus diskettes del DOS ponga la protección contra escritura para evitar la pérdida de información.

El disco interno, revestido de material magnético, se subdivide de la siguiente forma:

— Cada disco tiene dos caras, aunque no siempre se utilizan las dos.

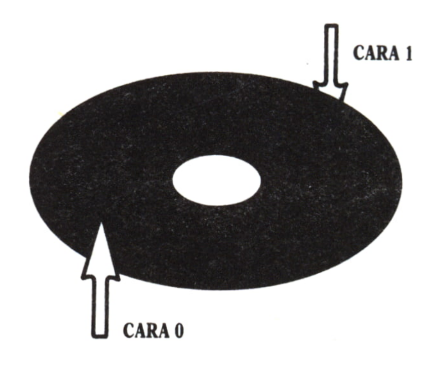


Fig. 3.

— Coronas circulares concéntricas, denominadas pistas; hay 40 pistas en un diskette estándar y 80 en los de alta capacidad.

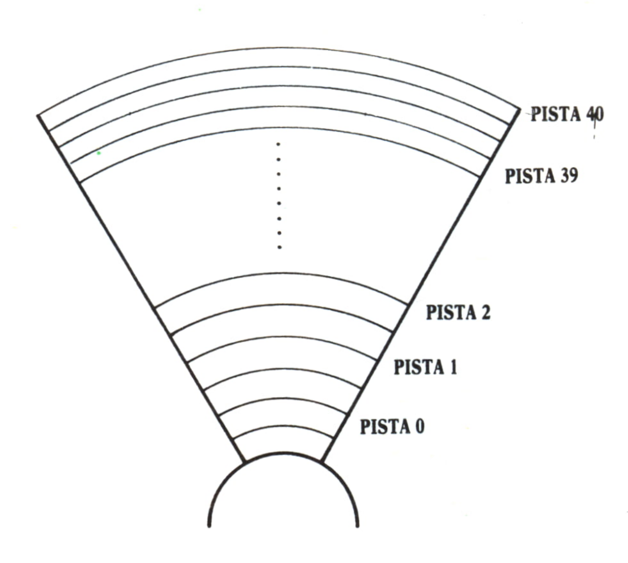


Fig. 4.

— Cada pista se subdivide en áreas más pequeñas denominadas sectores. En general, se cuenta con 9 sectores. Los de alta capacidad tienen 15 sectores.

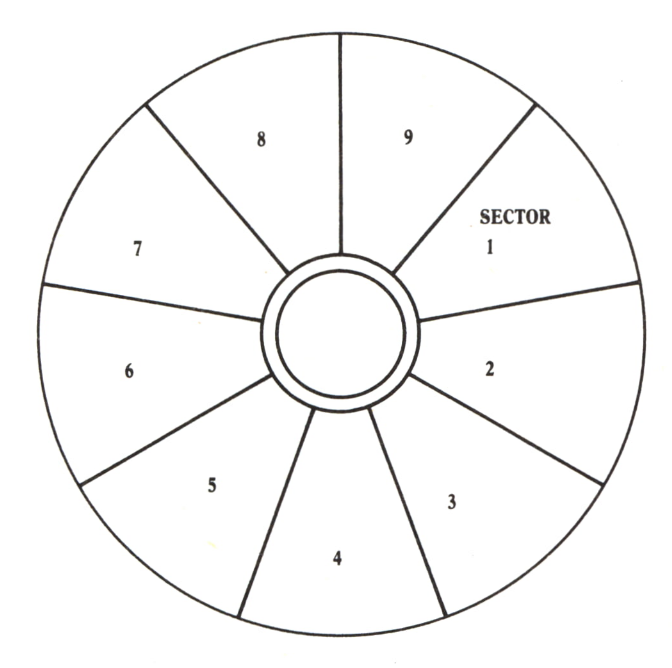


Fig. 5.

Las pistas de un diskette estándar se numeran del 0 al 39; los sectores, del 1 al 9, haciendo un total de 360 divisiones (40 pistas por 9 sectores) en cada cara (cara 0 y cara 1).

Como cada sector de una pista puede almacenar 512 bytes de información (aunque sean de distinto tamaño, todos pueden almacenar la misma cantidad de información). Por tanto, un diskette de dos caras puede almacenar 360K (36840 bytes), y si es de una cara, sólo la mitad, 180K.

Los de alta capacidad pueden almacenar hasta 1,2 Megabytes en sus 80 pistas y 15 sectores.



DISCO DURO (HARD DISK)

El disco duro más popular es el Winchester, desarrollado por IBM y que forma parte de gran cantidad de ordenadores personales.

Dependiendo de cuál sea, un disco duro puede almacenar entre 10 y 20 millones de bytes, teniendo un tiempo de acceso mucho menor que el de un diskette.

Se trata de un disco fijo que no se extrae de la unidad, estando constituido en general por varios discos superpuestos, que en lugar de ser plástico revestido de material magnético, como en los diskettes, es un metal revestido que da más rigidez al sistema.

El disco duro más común está formado por cuatro discos, cada uno de ellos con 306 pistas y 17 sectores que da una capacidad de 10 Mbytes. El del IBM PC AT es de una distribución similar, pero su capacidad es de 20 Mbytes.

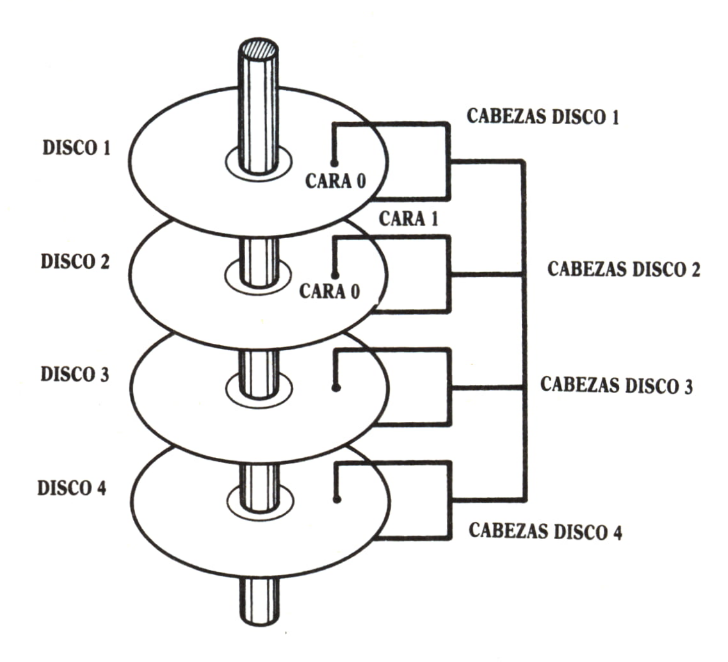


Fig. 6. Estructura de un disco duro.

Las principales precauciones que deben tomarse con un disco duro no se deben, evidentemente, a su manipulación, sin extendernos mucho, podemos resumirlas en las siguientes:

— Cuidado con el comando format, podemos perder toda la información del disco. Recomendamos utilizarlo siempre con el manual delante o redefinir dicho comando para evitar este problema.
— Hacer copias de seguridad en diskettes con el comando BACKUP.
— Recuperar los archivos con el comando RESTORE.
— Establecer un cuidadoso sistema de directorios.



UNIDADES DE DISCO

Como hemos visto, los ordenadores personales utilizan dos tipos de disco, los discos flexibles, denominados floppies o diskettes, que se pueden poner y quitar de la unidad de discos, y otro tipo que está siempre montado en su unidad, llamado, por esa razón, disco fijo o duro y que puede almacenar entre 30 y 60 veces lo que un diskette a mayor velocidad.

Si poseemos unidad de disco duro ésta recibirá el nombre de unidad de disco C. La configuración típica añade al disco duro una unidad de diskette.

Las unidades de disco flexible se denominan comenzando por la letra A. El caso más general es disponer de una o dos unidades de diskette que reciben el nombre de unidad A, la que se sitúa más a la izquierda o encima, y unidad B, la que se encuentra más a la derecha o debajo. La unidad principal en ausencia de disco duro es la A.

Podemos distinguir en ellas tres secciones principales: una que sujeta el floppy y lo hace girar, una que mueve y controla la cebeza lectora/escritora (o cabezas) de la unidad de disco y otra para comunicarse con el ordenador.

Cuando se inserta un diskette y se cierra la portezuela de la unidad, dos engranajes, uno por cada cara, encajan en el agujero central del disco. El engranaje inferior está conectado por una correa a un motor que hace girar el diskette en el interior de su funda protectora a una velocidad de 300 r.p.m.

En el interior de la unidad de disco también existen dos brazos móviles en los que se encuentran las cabezas que son muy pequeñas y de material cerámico. Las unidades de discos de simple cara sólo tienen una cabeza al final del brazo, las de dos caras tienen dos cabezas al final del mismo.

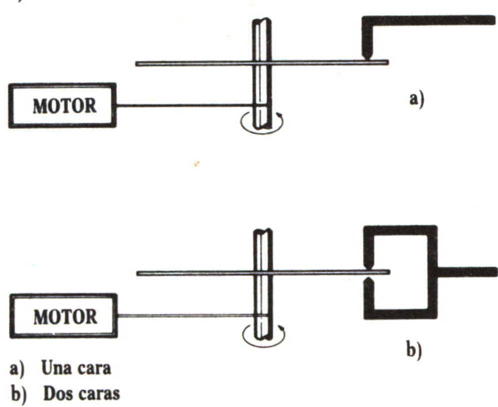


Fig. 7.

Con esto ya tenemos una idea del funcionamiento de nuestras unidades de disco. En el caso del disco duro el sistema es muy similar, la diferencia principal consiste en el hecho de que existen varios discos y, por tanto, varios grupos de cabezas (ver fig. 6).



ESPECIFICACION DE LA UNIDAD DE DISCO

Cuando se busca un fichero, el DOS debe saber en qué unidad debe buscar. En principio, el DOS busca en la unidad activa, es decir, la que señala el prompt cuando le especificamos el archivo que deseamos:

A>CARTA1.TXT

Buscará en la unidad de disco A.

Si queremos que lo busque en la unidad B, que es en la que hemos introducido el diskette correspondiente, tenemos dos posibilidades:

1. Cambiar la unidad de disco activa:

A>B:

Buscar ahora en la unidad B, al igual que habíamos hecho antes:

B>CARTA1.TXT

2. Indicarle que busque directamente en la unidad B, sin cambiar de unidad activa.

A>B:CARTA1.TXT

O también:

C>B:CARTA1.TXT



PERIFERICOS

Los periféricos son las partes del ordenador que nos ayudan a comunicarnos con él, y a el, con el mundo exterior. Como ya se dijo son principalmente:

— El teclado.
— El monitor.
— La impresora.
— El modem.
— Etcétera.

Cada uno de estos dispositivos utiliza una conexión para comunicarse con el ordenador que se llama INTERFACE (léase «interfeis»). Algunos de

los interfaces están incluidos dentro de la carcasa del ordenador, por ejemplo, el del teclado. Sin embargo, otros necesitan una terjeta especial para poder ser utilizados, que no es otra cosa que su interface.

Existen dos tipos de interfaces que dependen del modo de comunicación con el ordenador:

Interface paralelo. Transmiten el byte correspondiente de una vez. Necesitan, por tanto, un hilo de transmisión para cada bit, además de otras líneas eléctricas. En general, las impresoras utilizan este sistema.

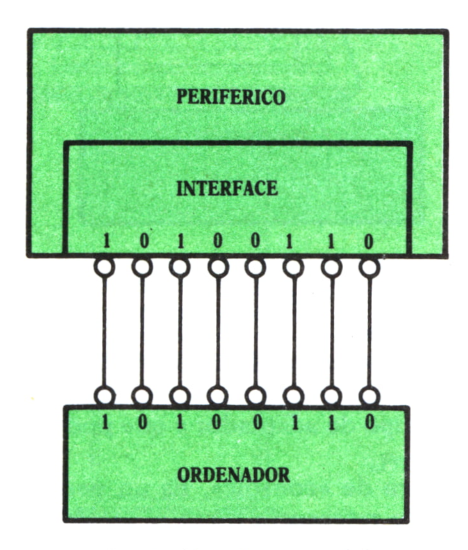


Fig. 8. Interface paralelo.

Interface serie. Los ocho bits que forman el byte se envían de uno en uno, la transmisión es más lenta, pero sólo es necesario un hilo y las líneas eléctricas. Ejemplos de este sistema son: el teclado y el modem.

El monitor es una excepción. Utiliza una memoria mapeada que lo hace más rápido, debido a la mayor velocidad de movimiento de bytes en memorias que entre dispositivos.

Las unidades de disco utilizan un sistema serie, transmitiendo un bit cada vez que son agrupados en forma de bytes, y éstos, a su vez, en bloques (recordamos que estos periféricos manejan bloques de datos), estas acciones son tan rápidas que el usuario no se percata del trabajo que están realizando los adaptadores y el disco duro.

Podemos decir, por tanto, que el ordenador ve los periféricos desde dos puntos de vista:

— Periféricos con interface serie o paralelo.
— Periféricos orientados al manejo de caracteres o bloques de datos.

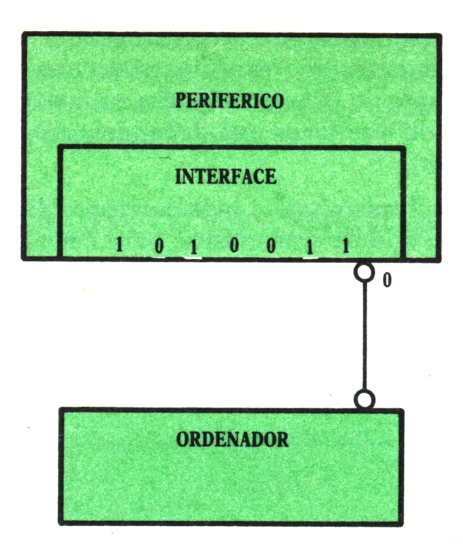


Fig. 9. Interface serie.

El software utilizado para el manejo de los periféricos se encuentra, como ya dijimos, en el BIOS del ordenador. Para poder trabajar cada dispositivo, al igual que sucedía con los archivos, recibe un nombre que normalmente se basa en el tipo de adaptador y que podrá ser utilizado con los comandos del DOS:

CON es la abreviatura de «console». Es un periférico tanto de salida como de entrada, y hace referencia tanto al teclado (que es de entrada) como a la pantalla (que es de salida). El DOS sabe en cada caso cuál se utiliza.

PRN es la abreviatura de «printer». Es un periférico de salida, y hace referencia a la impresora en paralelo que el DOS utiliza, salvo que se le indique otra cosa (es análogo a lo que hace el DOS con la unidad de disco por defecto). Pueden conectarse hasta tres impresoras en paralelo denominadas LPT1, LPT2 y LPT3; el DOS supone que PRN significa LPT1 salvo que se le indique lo contrario.

AUX es la abreviatura de «auxiliary». Es un periférico tanto de entrada como de salida, y hace referencia al puerto de comunicaciones que el DOS utiliza, en tanto no se indique lo contrario. Se pueden conectar uno o dos puertos de comunicaciones denominados COM1 y COM2; en principio, el DOS supone que AUX significa COM1. En un sistema típico, COM1 se podría emplear para un modem; y COM2 se podría emplear para una impresora serie.

Las reglas para la utilización de los nombres son las siguientes:

1. Se pueden utilizar en el mismo lugar que los nombres de archivo.

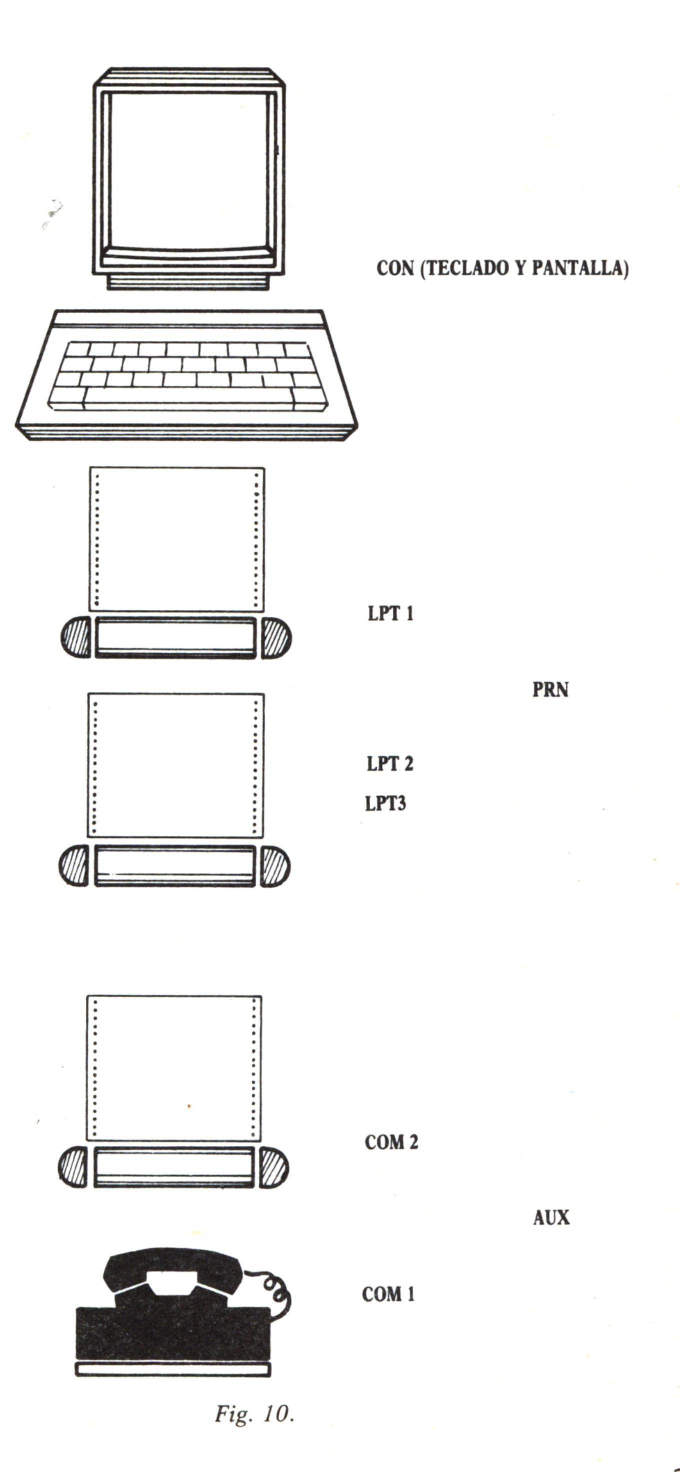


Fig. 10.

2. Opcionalmente pueden ir seguidos de dos puntos.
3. Si llevan nombre de extensión el DOS la ignora.
4. No se debe utilizar el nombre de un dispositivo que no exista. Por ejemplo, no utilizar LPT2 si sólo disponemos de una impresora.
5. No utilizar los nombres de los periféricos para ningún archivo.



PRIMEROS COMANDOS

Vamos a aprender algunos comandos que nos permitirán manejar ficheros, discos y periféricos.



DIR

Ya hemos mencionado el directorio en las páginas de este libro, siendo éste el momento oportuno de aclararle cuanto necesita saber en torno al mismo.

El directorio es una lista de cada uno de los archivos que se guardan en el disco que se está utilizando en ese momento y que es conservada y actualizada por el DOS. Si crea y guarda un nuevo archivo, el DOS lo añade a la lista. Si revisa un archivo ya existente, el DOS toma nota de ello también. El directorio hace inútil la necesidad de mantener un registro aparte de cada archivo que se guarda en un disco. Se puede pedir al DOS que muestre el directorio en cualquier momento en que el DOS esté mostrando el prompt. Este directorio constará de lo siguiente:

— Nombre del archivo. Como ya se sabe, son hasta 8 caracteres y opcionalmente un punto y una extensión de tres caracteres.
— Tamaño del archivo en bytes. Aprovechamos la ocasión para aclarar que un byte es la cantidad de espacio de memoria requerido para guardar un carácter en la memoria del ordenador o en un disco.
— Fecha y hora en las que el archivo fue creado o se realizó la última modificación.
— Nombre del disco (es opcional).
— Número de archivos almacenado en el disco.
— Número total de bytes ocupados del disco.

Para extraer el directorio de un disco de la unidad activa basta con escribir DIR, que es el nombre del comando del directorio.

Con el disco del DOS en la unidad A, y siendo ésta la unidad activa, escribimos DIR y pulsamos [Intro].

El DOS muestra una lista de los archivos que están contenidos en el diskette del DOS. Como el directorio es mayor de 25 líneas, parte del directorio sube por la pantalla y desaparece por arriba. Para remediar esto y po-

```
Mie 29/04/1987 10:09:52,24 A>DIR

 Volumen en unidad A etiquetado DOS V300(1)
 Directorio de  A:\

COMMAND  COM     22698  18/09/84  12:00
CONFIG   SYS        52   4/03/86  20:31
001-DOS3 LBL         0   5/04/87   1:40
ANSI     SYS      1641  18/09/84  12:00
SORT     EXE      1648  18/09/84  12:00
SHARE    EXE      8576  18/09/84  12:00
FIND     EXE      6392  18/09/84  12:00
ATTRIB   EXE     15123  18/09/84  12:00
MORE     COM       286  18/09/84  12:00
ASSIGN   COM       995  18/09/84  12:00
PRINT    COM      8099  18/09/84  12:00
SYS      COM      3741  18/09/84  12:00
CHKDSK   COM      9675  18/09/84  12:00
FORMAT   COM      9220  18/09/84  12:00
VDISK    SYS      3154  18/09/84  12:00
BASIC    COM     17024  18/09/84  12:00
BASICA   COM     26880  18/09/84  12:00
FDISK    COM      8252  18/09/84  12:00
COMP     COM      3632  18/09/84  12:00
TREE     COM      2522  18/09/84  12:00
BACKUP   COM      5700  18/09/84  12:00
RESTORE  COM      5669  18/09/84  12:00
LABEL    COM      1309  18/09/84  12:00
DISKCOPY COM      4240  18/09/84  12:00
DISKCOMP COM      3843  18/09/84  12:00
KEYBSP   COM      2073  18/09/84  12:00
HBASIC   EXE      8142   7/06/85
HGC      COM      1591  20/08/85
FONT     COM      2317   1/11/83
MODE     COM      5354  18/09/84  12:00
SELECT   COM      2106  18/09/84  12:00
GRAPHICS COM      3111  18/09/84  12:00
RECOVER  COM      4210  18/09/84  12:00
EDLIN    COM      7343  18/09/84  12:00
GRAFTABL COM      1190  18/09/84  12:00
FC       COM     16408  12/12/85   9:43
AUTOEXEC BAT        72   9/07/86  13:33
EXE2BIN  EXE      2816  18/09/84  12:00
LINK     EXE     45970  18/09/84  12:00
DEBUG    COM     15334  18/09/84  12:00
PRN60132 BAS        62   1/05/86  17:53
       41 Archivo(s)     11264 bytes libres
```

Fig. 11.

der ver el directorio completo debemos conseguir que se detenga la ejecución del comando. Tenemos los siguientes métodos:

— Presionar simultáneamente las teclas [Control] y [Num Lock]. Se interrumpe cualquier actividad del sistema hasta que se vuelve a pulsar otra tecla.

— Presionar simultáneamente [Control] y [S]. El resultado será equivalente al anterior.

— Presionar simultáneamente las teclas [Control] y [Break]. Termina cualquier actividad del sistema, con lo cual no podremos terminar de ver el directorio, pero puede ser útil si sólo deseamos ver una parte del mismo.

— Presionar simultáneamente las teclas [Control] y [Scroll Lock]. El resultado es equivalente al anterior.

— Presionar simultáneamente [Control] y [C]. E! resultado será equivalente al anterior.

— Teclear el comando DIR y a continuación /P, con ello conseguiremos que escriba sólo las 23 líneas que caben en la pantalla y espere a que pulsemos una tecla para continuar.

— Teclear DIR/W, con ello aparecen el nombre de cinco archivos en cada línea, aunque no aparecen sus extensiones.

```
  Volumen en unidad A sin etiqueta
  Directorio de  A:\

COMMAND  COM   CONFIG   SYS   ANSI     SYS   ASSIGN   COM   ATTRIB   EXE
BACKUP   COM   LINK     EXE   VDISK    DOC   CHKDSK   COM   COMP     COM
DISKCOMP COM   DISKCOPY COM   EDLIN    COM   FDISK    COM   FIND     EXE
FORMAT   COM   GRAFTABL COM   GRAPHICS COM   JOIN     EXE   KEYBFR   COM
KEYBGR   COM   KEYBIT   COM   KEYBSP   COM   KEYBUK   COM   LABEL    COM
MODE     COM   MORE     COM   PRINT    COM   RECOVER  COM   RESTORE  COM
SELECT   COM   SHARE    EXE   SORT     EXE   SUBST    EXE   SYS      COM
TREE     COM   VDISK    SYS   DEBUG    COM   EXE2BIN  EXE   RELOJ    COM
AUTOEXEC BAT
       41 Archivo(s)     38912 bytes libres
```

Fig. 12.

Si desea imprimir y obtener por pantalla al mismo tiempo, puede utilizar la orden [Control] y [Prt Sc]. Esto le indica al DOS que comience a imprimir cualquier cosa que salga por la pantalla hasta que se pulse de nuevo [Control] [Prt Sc]. Para probar esta función debe asegurarse que la impresora esté encendida, déle la orden de impresión simultánea y escriba DIR.

IMPRESORA ENCENDIDA
[Control] [Prt Sc]
A>dir

El DOS sacará el directorio del disco del sistema, pero esta vez cada línea se imprime mientras va saliendo por pantalla. El directorio sale más

lentamente, porque el DOS espera a que la línea se termine de imprimir para pasar a la siguiente.

Si no desea que en la impresión salga el comando DIR, basta con alterar el orden de los comandos:

IMPRESORA ENCENDIDA
A>dir
[Control] [Prt Sc]

Ahora la impresión comienza con la primera línea del directorio; el comando DIR no se imprime.

Si consideramos ahora la posibilidad de necesitar el directorio de otra unidad, observamos las siguientes soluciones:

— Indicarle el nombre de la unidad de disco de la que queremos el directorio. Por ejemplo:

A>DIR B:

Nos permite obtener el directorio de la unidad B sin cambiar a ella.

— Cambiar de unidad, y una vez en la unidad deseada pedir el directorio.

A>B:
B>DIR

Podemos indagar más aún en este comando: deseamos ahora encontrar un fichero cuyo nombre es MSDOS1.LIB, para ello al pedir el directorio le indicamos el nombre del fichero:

A>DIR MSDOS1.LIB

```
A>DIR MSDOS1.LIB

 Volumen en unidad A sin etiqueta
 Directorio de  A:\

No se encontraron archivos
```

Fig. 13.

O bien, si queremos buscarlo en la unidad B, también:

A>DIR B:MSDOS1.LIB

```
A>DIR B:MSDOS1.LIB

 Volumen en unidad B sin etiqueta
 Directorio de  B:\

MSDOS1    LIB     8704   9/04/87  15:36
        1 Archivo(s)     206848 bytes libres
```

Fig. 14.

Abrimos más posibilidades con los caracteres comodín.



¿QUE SON LOS CARACTERES COMODIN?

Para hacer más fácil el manejo de los archivos de disco, la mayoría de los comandos que actúan sobre los archivos permiten utilizar caracteres comodín para poder manejar varios archivos al mismo tiempo. De esta manera, cuando se desea hacer una cosa con varios archivos no es necesario emplear un comando para cada archivo. Los caracteres comodín pueden representar cualquier otro carácter en un nombre de archivo. Existen dos caracteres comodín:

El asterisco *
La interrogación ?

1. El asterisco hace fácil llevar a cabo comandos sobre grupos de archivos con extensiones similares; este carácter puede representar hasta los ocho caracteres del nombre de un archivo o hasta los tres caracteres de la extensión, pudiendo ser también uno solo de ellos.

Proponemos los siguientes ejemplos para su comprensión, que serán realizados con el comando DIR, por ser el que conocemos.

— El índice y los capítulos de este libro se han ido almacenando en los ficheros MSDOSIND.LIB, MSDOS1.LIB, MSDOS2.LIB, MSDOS3.LIB,...

Las figuras de cada capítulo llevan el mismo nombre, pero su extensión es .FIG.

Vamos a pedir en primer lugar el directorio de todos ellos, para esto, como poseen una parte del nombre común utilizamos el *, tanto para los finales de los nombres como para su extensión:

A>DIR B:MSDOS*.*

```
A>DIR B:MSDOS*.*

 Volumen en unidad B sin etiqueta
 Directorio de  B:\

MSDOS2    BAK     14976    9/04/87    0:28
MSDOS1    BAK      8320    8/04/87   22:12
MSDOS2    LIB     15232    9/04/87   15:48
MSDOSIND BAK        384    8/04/87   12:00
MSDOSA    FIG       128   19/04/00   10:55
MSDOS3B   LIB      6528   13/04/00   13:35
MSDOSIND LIB       1280   13/04/00   13:49
MSDOS1    LIB      8704    9/04/87   15:36
MSDOS3    BAK     40960   13/04/00   13:41
MSDOS3B   BAK      4864   12/04/00   20:16
MSDOSAP1 LIB       1152   13/04/00   13:58
MSDOS4    LIB       128   14/04/00   12:44
MSDOS3    LIB     40960   19/04/00   10:52
MSDOSB    FIG       128   19/04/00   10:55
MSDOS1    FIG       128   19/04/00   10:56
MSDOS2    FIG       128   19/04/00   10:56
MSDOS3    FIG       128   19/04/00   10:56
MSDOS4    FIG       128   19/04/00   10:57
        18 Archivo(s)     206848 bytes libres
```

Fig. 15.

Que es equivalente a **DIR B:MSDOS***, ya que al no especificar ninguna extensión, el **DOS** dará todos los archivos que empiecen por **MSDOS**.

Obtengamos ahora sólo los que corresponden a capítulos de libro:

A>DIR B:*.LIB

```
A>DIR B:*.LIB

 Volumen en unidad B sin etiqueta
 Directorio de  B:\

MSDOS2    LIB     15232    9/04/87   15:48
MSDOS3B   LIB      6528   13/04/00   13:35
MSDOSIND LIB       1280   13/04/00   13:49
MSDOS1    LIB      8704    9/04/87   15:36
MSDOSAP1 LIB       1152   13/04/00   13:58
MSDOS4    LIB       128   14/04/00   12:44
MSDOS3    LIB     40960   19/04/00   10:52
         7 Archivo(s)     206848 bytes libres
```

Fig. 16.

2. La interrogación sustituye únicamente a un carácter en un nombre de archivo o en una extensión. Es más frecuente utilizar el asterisco.

```
A>DIR DIR B:MSDOS1?

 Volumen en unidad B sin etiqueta
 Directorio de  B:\

MSDOS1   BAK     8320   8/04/87  22:12
MSDOS1   LIB     8704   9/04/87  15:36
MSDOS1   FIG      128  19/04/00  10:56
        3 Archivo(s)    206848 bytes libres
```

Fig. 17.



FORMAT

El comando **FORMAT** prepara un diskette nuevo para su utilización posterior. Si se trata de un diskette ya utilizado, el formateo borrará cualquier archivo existente; por ello debe estar seguro de no formatear un diskette que contenga archivos necesarios.

El **DOS** va a comprobar los defectos de la superficie magnética del diskette, y marca todos los sectores que están en malas condiciones para que no se utilicen. Al terminar el formateo, el **DOS** saca un mensaje que indica el número máximo de bytes que el diskette podrá almacenar, cuántos bytes (si hay) están inutilizables y cuántos bytes son aprovechables para almacenar archivos.

```
A>format b:
Inserte un nuevo diskette en la unidad B:
y pulse Intro para continuar

Formateando...Formateo completo

    362496 bytes es el espacio total en disco
    362496 bytes disponibles en disco

Formatear otro diskette (S/N)?n
A>dir b:

 Volumen en unidad B sin etiqueta
 Directorio de  B:\

No se encontraron archivos
```

Fig. 18.

El DOS reconoce si las unidades son de una o dos caras y formatea de acuerdo a ello. Sin embargo, se le puede especificar el tipo de formato que deseamos.

Automáticamente formatea con nueve sectores por pista.

La forma de utilizar este comando es la siguiente:

— Si posee dos unidades de disco:

Disco del DOS en la unidad A
A>format b:
A>format b:

— Si posee disco duro:

Disco a formatear en la unidad A
C>format a:

¡Cuidado! Si no especifica la unidad, y la unidad por defecto contiene el diskette del sistema, o lo que es peor, se trata del disco duro, podría perder todo el contenido del mismo. Tome precauciones antes de pulsar la tecla [Intro] para asegurarse que la unidad indicada es la correcta.

El comando **FORMAT** permite una serie de parámetros:

/V permite ponerle nombre al dikette.

```
A>format b:/v
Inserte un nuevo diskette en la unidad B:
y pulse Intro para continuar

Formateando...Formateo completo

Etiqueta de Volumen (máximo 11 caracteres y/o INTRO)LIBRO MSDOS


    362496 bytes es el espacio total en disco
    362496 bytes disponibles en disco

Formatear otro diskette (S/N)?N
A>DIR B:

 Volumen en unidad B etiquetado LIBRO MSDOS
 Directorio de  B:\

No se encontraron archivos
```

Fig. 19.

/1 formatea sólo una cara del diskette.

/4 formatea nueve sectores por pista en un diskette de dos caras colocado en una unidad de alta capacidad.

/8 formatea el diskette con 8 sectores por pista.

/S realiza, además, una copia del sistema operativo en el diskette, de forma que se pueda arrancar con él el sistema. Esto puede resultar muy útil para discos de trabajo si no se dispone de unidad de disco duro, ya que no será necesario utilizar el diskette del DOS, bastará con introducir directamente el disco de trabajo.

Si obtiene algún error en el disco durante el formateo, inténtelo de nuevo, en ocasiones un nuevo formato recupera el disco que pensábamos que estaba en malas condiciones.



COPY

El comando COPY es uno de los comandos más versátiles del DOS. Copia archivos entre unidades de disco u otros periféricos, con la posibilidad de cambiarle el nombre al archivo. Puede concatenar dos o más archivos en otro archivo, o añadir archivos a otro. Sus expresiones generales son:

Para copiar un archivo:

COPY/A/B unidad de disco origen:nombre archivo/A/B unidad de disco destino:nuevo nombre/A/B/V

Para concatenar:

COPY/a/b unidad de disco1:nombre1/A/B + unidad disco2: nombre2/A/B +... unidad disco0:nombre0/A/B/V



COPIANDO CON COPY

En primer lugar vamos a ver su posibilidad de copiar archivos entre unidades de disco o entre unidades de disco y un dispositivo. COPY utiliza un archivo como si fuera un dispositivo o periférico, de forma que logra, además de la copia entre unidades de disco, copia desde unidades de disco a un dispositivo, desde un dispositivo a un archivo de disco, o desde un dispositivo a otro.

El comando tiene dos posibilidades:

/A para copiar un fichero ASCII. Se utiliza en archivos de texto que se caracterizan por la utilización de un carácter especial para determinar el final del fichero. Este carácter se representa por Ctrl-Z, CHR$(26), o 1A en hexadecimal. Cualquier programa que trabaje con ficheros de texto reconocerá este carácter como final del fichero, mientras que uno que no sea de texto lo tomará como un dato más o como una posición de memoria más. Cuando copiamos de un dispositivo que no sea la unidad de disco, el DOS asume que se trata de formato ASCII.

/B para copiar un fichero en binario. Si el fichero del que queremos copiar tiene el carácter Ctrl-Z al final, lo ignora, y si se trata del fichero destino, no le añade el Ctrl-Z. Cuando copiamos de disco a disco el DOS asume el formato binario de los datos.

Existe una tercera posibilidad de COPY: /V, se utiliza para verificar que la copia se ha realizado correctamente y puede utilizarse al mismo tiempo que los otros dos.



CONCATENANDO CON COPY

Nos fijamos en la estructura que debe tener el comando para utilizarlo con esta finalidad:

/A, /B y /V realizan la misma función que en el apartado anterior. Los primeros archivos que se ponen son los archivos origen, y son aquellos que quieres unir. Los archivos destino pueden ser más de uno, y se sitúan al final sin ponerles signo +.

Un ejemplo de utilización sería el siguiente:

A>COPY A:CURSO1.TXT + B:CURSO2.TXT + B:CURSO3.TXT A: CURSO.87/V

El contenido de CURSO1.TXT es trasladado a CURSO.87, y los contenidos de CURSO2.TXT y CURSO3.TXT son añadidos a CURSO.87. /V se encarga de verificar que la concatenación se ha realizado correctamente.

A>COPY *.LIB + *.FIG *.TOT

Tomaría los archivos con su correspondiente nombre y la extensión .LIB y lo uniría con el archivo del mismo nombre y extensión .FIG, situando el resultado en el archivo del mismo nombre y extensión .TOT.

Si recordamos el ejemplo puesto con estos nombres, en los que teníamos los capítulos de un libro con la extensión .LIB y el nombre MSDOSn, donde n era el número del capítulo, las figuras de cada capítulo se almacenaban en ficheros del mismo nombre y extensión .FIG, tendríamos, por tanto, como resultado, archivos, en los que estaría el texto y las figuras de cada capítulo juntas:

MSDOS1.LIB + MSDOS1.FIG → MSDOS1.TOT
MSDOS2.LIB + MSDOS2.FIG → MSDOS2.TOT
MSDOS3.LIB + MSDOS3.FIG → MSDOS3.TOT
MSDOS4.LIB + MSDOS4.FIG → MSDOS4.TOT

Podemos añadir un ejemplo más que nos permitirá la concatenación de todos los archivos del libro en uno solo:

A>COPY *.TOT MSDOS.TOT.

También podremos de esta manera cambiar la fecha y hora de almacenamiento de los archivos de un disco:

B>COPY A:MSDOS.TOT +,,A:

Las comas indican al DOS que no hay ningún fichero para añadir al MSDOS.TOT, entonces lo volverá a poner en la unidad indicada, pero cambiando la fecha.



ENVIANDO ARCHIVOS A LOS PERIFERICOS CON COPY

Podemos enviar una copia de un archivo a cualquier periférico de salida. Por ejemplo, puede copiarse un archivo a un puerto de comunicaciones y se envía, por tanto, a cualquier dispositivo que esté unido a él; éste sería el caso de una línea de telecomunicaciones que se encuentre conectada con otro ordenador.

El comando COPY adquiere la siguiente estructura:

COPY (nombre del archivo)(periférico)

o

COPY (periférico)(nombre de archivo)

Debe asegurarse que el periférico existe o está conectado, ya que si no existe o no está listo el DOS puede quedar bloqueado.

Como ejemplo de utilización vamos a crear un fichero de texto, enviando caracteres desde el periférico del teclado y monitor (CON) al disco y, una vez en él, lo enviaremos desde el disco a la impresora (PRN).

Para la primera parte tecleamos:

COPY CON B:SALUDO.TXT

Todo lo que se teclee será enviado al fichero indicado (SALUDO.TXT) hasta que el DOS encuentre el carácter indicador de final de fichero de texto (Ctrl-Z).

Supongamos el siguiente texto:

Hola a todos:

Os envío un saludo desde el ordenador.
Como no quiero extenderme mucho,
me despido hasta otra ocasión.

No olvide que para terminar una línea debe pulsar [Intro], si quiere una línea en blanco dos veces [Intro], y al final el carácter ya mencionado (se encuentra directamente en la tecla de función F6 de su ordenador).

Verifique ahora con el directorio que el fichero SALUDO.TXT se encuentra en el disco situado en la unidad B.

A>dir b: o   B>dir

Más aún, puede volverlo a enviar a la pantalla del ordenador (no va a perder el contenido) mediante:

A>COPY B:SALUDO.TXT CON

Recordamos que el DOS sabía cuándo CON (consola) actúa como periférico de entrada (teclado), y cuándo como periférico de salida (monitor).

Obtenemos, por último, una impresión del fichero con:

A>COPY B:SALUDO.TXT PRN



TYPE

Continuando con el ejemplo anterior, teclee lo siguiente:

TYPE SALUDO.TXT

```
TYPE SALUDO.TXT
Hola a todos:
Os envio un saludo desde el ordenador
Como no quiero extenderme mucho.
me despido hasta otra ocasión.

                STM
```

Fig. 20.

Observará que el DOS rápidamente extrae todas las líneas y vuelve a poner el prompt, quedando a su disposición.

Ya se habrá dado cuenta de que es equivalente a:

COPY SALUDO.TXT CON

Sin embargo, el comando TYPE es la manera más cómoda y rápida de ver lo que hay en un archivo. Evidentemente será útil en el caso de archivos de texto, ya que, en caso contrario, no encontraríamos nada legible. Compruébelo con los siguientes ejemplos:

```
B>TYPE PAL.BAS
 +

4L
à "LONGITUD DE LA PALABRA":LY
à P(Lθ)e
(ë A$┬""w
2é I┬ ├ (Lθ)Ç
<à C$A
Fë A$┬A$θC$┬
Pï C$┬"A" ∩ C$┬"E" ∩ C$┬"I" ∩ C$┬"O" ∩ C$┬"U" — ë P(I)┬:ë ┌┐≡
```

Fig. 21.

```
TYPE ALFAC.EXE
MZ≡
     ü┬
┌√ü
```

Fig. 22.

```
B>TYPE A:WS.COM
θ
Eθ┌Dθ2OθE  Copyright (C) 1979, 1984, MicroPro International Corp.   8086/8088 WordStar
serial # WS9999XX  Release 3.40-4.0       IBM Personal Computer        IBM Parallel Printer       No communications protoc
ol           Parallel printer driver
```

Fig. 23.

Siempre que los archivos sean de texto, puede interesar revisarlos, sin necesidad de una copia impresa. Tenga en cuenta que si el archivo es más largo que una pantalla, puede interrumpir la pantalla pulsando Ctrl-Num Lock.

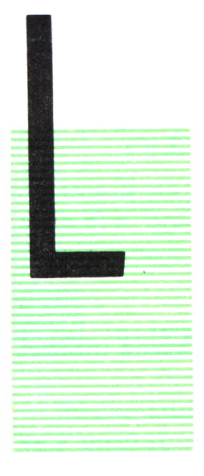
Quedan muchos comandos por ver, sin embargo, consideramos que en principio para aprender y conocer el DOS no son necesarios más que los mostrados en este capítulo.

No obstante, el capítulo 5 estará dedicado a mostrarles una relación de los comandos del DOS y sus funciones, como ya dijimos no es necesario estudiarlos, conviene saber que existen, en caso de necesitarlos lo mejor es acudir al manual.

PREPARACION DEL SISTEMA 4



INTRODUCCION



LA configuración de un sistema supone una preparación del mismo para que lleve a cabo la función que nosotros, los usuarios, deseamos. Existen varios niveles de configuración:

La configuración física

Corresponde a la instalación del sistema, la impresora, el monitor, el modem, requieren que se manipulen los switches (interruptores) internos del interior del ordenador.

La configuración de trabajo

Consiste en la organización requerida para los directorios de discos y diskettes.

La configuración del sistema operativo

Consiste en adaptar el BIOS a nuestras necesidades. La version 2 del DOS posee un fichero especial en el directorio: CONFIG.SYS. Su contenido son comandos que permiten la flexibilidad mencionada en la configuración de nuestro sistema. El DOS trata este fichero antes de AUTOEXEC.BAT.



COMANDOS DE CONFIGURACION

Son los comandos que están permitidos en el fichero de configuración. No se escriben a través del teclado, sino que están reservados para CONFIG.SYS, definiendo en él las características operativas del DOS.



BREAK

Su función es la misma que en el resto del sistema operativo. Si se activa (ON), el DOS chequea la posible presión de Ctrl-Break cada vez que un programa solicita la actividad del DOS. Si se desactiva (OFF), sólo se chequea cuando existe control sobre el monitor, teclado, impresora, o adaptador asíncrono serie.

Es decir, en un programa que realiçe muchos accesos a disco, y que sea de gran longitud, tendremos que activar el comando BREAK para que el programa pueda ser detenido en cualquier instante, sin esperar que el monitor o teclado reciban control.

La sintaxis de este comando es:

BREAK ON (lo activa)
BREAK OFF (lo desactiva)



BUFFERS

Este comando indica al DOS los buffers de disco que se van a utilizar. Su sintaxis es:

BUFFERS=nn

Donde nn es el número de buffers deseados entre 1 y 99. Inicialmente el DOS crea dos buffers. Aunque se pueden crear hasta 99, se recomienda no superar los 20, ya que el tiempo empleado por el DOS en chequearlos superaría el tiempo que ahorramos al crearlos. Un número típico de buffers utilizados es 10.

Un buffer de disco es un espacio de memoria RAM reservado por el DOS. Su dimensión es la misma que la del sector de un disco (normalmente 512 bytes). El DOS los utiliza para leer o grabar en un disco.



DEVICE

Su sintaxis es:

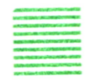
DEVICE=unidad de disco:nombre del archivo.extensión

Con este comando y el software adecuado podrá hacer un uso mejor de las posibilidades ofrecidas por el hardware.

En el archivo se encuentra el software necesario para que el DOS controle los periféricos.

Un ejemplo de archivo utilizado es ANSI.SYS, que es uno de los dispo-

sitivos controladores del DOS. ANSI son las reglas correspondientes al American National Standard Institute (Instituto Nacional de Normalización Americano). El archivo ANSI.SYS establece un conjunto de métodos normalizados de gestión de las visualizaciones, que controla la forma en que aparecen en la pantalla o se borran los caracteres, la manera en que se mueve el cursor o cómo se utilizan los colores. Hay ciertos programas para los cuales es necesaria la presencia de un archivo CONFIG.SYS que contenga el comando DEVICE=ANSI.SYS.



FILES

Su sintaxis es:

FILES=nn

Donde nn se encuentra entre 5 y 99, siendo el número inicial dado por el DOS 8, que suele ser suficiente. Determina el número de archivos que el DOS puede utilizar a un mismo tiempo.



SHELL

El comando SHELL designa a un programa (archivo de comandos) que va a sustituir al COMMAND.COM en su función de procesador de comandos del DOS.



COUNTRY

Permite establecer un código que indique al DOS las características de los formatos de un determinado país en fechas, símbolos monetarios y separadores decimales.

Su sintaxis es:

COUNTRY código

Donde código es el número de código asignado a cada país y suele coincidir con el prefijo telefónico internacional de cada nación. Hay que escribir los tres dígitos, aunque se trate de ceros a la izquierda.

Estados Unidos	001
Países Bajos	031
Bélgica	032
Francia	033
España	034

Italia	039
Suiza	041
Reino Unido	044
Dinamarca	045
Suecia	046
Noruega	047
Alemania	049
Australia	061
Finlandia	358
Israel	972

Si el ordenador que está utilizando fue construido en otro país puede ser necesario alterar el código.



CREANDO CONFIG.SYS

Como se trata de un archivo de texto, se puede realizar con un editor (por ejemplo, EDLIN) o utilizando el comando COPY, como se ha visto en el capítulo anterior.

Recuerde que debe llamarse CONFIG.SYS, para que el DOS pueda reconocerlo en el momento de arrancar el sistema.

```
BUFFERS=10
DEVICE=ANSI.SYS
COUNTRY=034
```

Coloque el archivo en el directorio del DOS. Si ha incluido la segunda línea, DEVICE=ANSI.SYS, copie ANSI.SYS en el disco que vaya a utilizar para inicializar el sistema.

Ya puede arrancar de nuevo el sistema para que el DOS lea el fichero CONFIG.SYS que acaba de crear.

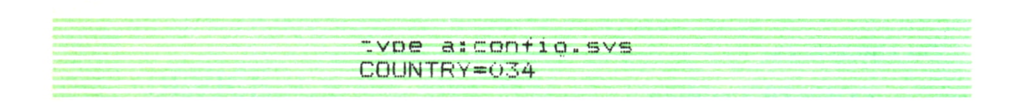


Fig. 1.



AUTOEXEC.BAT

Ya hemos hablado de este fichero al comentar el arranque del sistema. Se trata de un archivo batch en el que se pueden introducir todos los pro-

cedimientos de inicialización del sistema que se desee, y que se ejecutará automáticamente por el DOS cada vez que se activa el sistema.

Vamos a explicar brevemente qué es un archivo batch. El término batch (lote) tuvo su origen en los días de los grandes ordenadores, cuando la mayor parte del trabajo se realizaba dejando un paquete de tarjetas perforadas al departamento de proceso de datos. Las tarjetas perforadas debían contener toda la información necesaria para el perfecto funcionamiento del programa, ya que no había posibilidad de interacción con el sistema. El personal de proceso de datos realizaba estas tareas por lotes y entregaba los resultados.

Cuando se utiliza un comando batch se realiza lo mismo, porque el comando batch contiene todas las instrucciones necesarias para ejecutar una tarea. En la actualidad, batch se utiliza para denominar una tarea del ordenador que se realiza sin ninguna interrupción, frente a las tareas interactivas en las que el usuario debe resolver la situación en cada momento.

Como ya dijimos, el DOS pone a los comandos batch la extensión .BAT. la diferencia entre AUTOEXEC.BAT y cualquier otro archivo del DOS que sea batch es que AUTOEXEC se ejecuta cada vez que se inicializa el sistema.



REGLAS PARA AUTOEXEC.BAT

— Lógicamente el archivo deberá llamarse AUTOEXEC.BAT para que el DOS lo localice en el momento de arrancar, y no debe coincidir con el nombre de ningún otro archivo del disco. Lo situaremos en el disco del DOS con el que inicializamos el sistema.

— Por ser un archivo batch debe contener texto en ASCII. Para crearlo podremos utilizar el EDLIN, el comando COPY o cualquier otro editor de textos, teniendo en cuenta que si utilizamos un procesador debemos encontrarnos en el modo de edición de programas o en el modo NO-DOCUMENTO.

— Si AUTOEXEC.BAT se ejecuta después de arrancar el DOS, la fecha y la hora no serán actualizados, por tanto, deberán ponerse los comandos DATE y TIME entre los comandos de AUTOEXEC.BAT.

Veamos algunos ejemplos de archivos AUTOEXEC.BAT:

```
B>type autoexec.bat
reloj/s
ecommand ecomglo.set
hgc full print
```

Fig. 2.

```
A>type autoexec.bat
keybsp
reloj/s
```

Fig. 3.

```
A>type autoexec.bat
ECHO OFF
CLS
SPKBD
VER
PATH=A:\:B:
PROMPT $N$G
WS
```

Fig. 4.

```
A>type autoexec.bat
echo off
keybsp
cls
ngc full save print
font
date
time
ver
menu
```

Fig. 5.

Como vemos podemos utilizar cualquier comando permitido por el DOS, así como los subcomandos batch y parámetros.



SUBCOMANDOS BATCH

ECHO

Permite conectar o desconectar el monitor mientras se realizan los comandos batch. Permite, además, la visualización de un mensaje en la pantalla, aunque no esté activo.

PAUSE

Detiene el proceso hasta que se presione una tecla, permitiendo, además, que se muestre un mensaje (siempre que ECHO esté ON).

GOTO

Produce un salto a la línea inmediatamente posterior a la etiqueta indicada. Las etiquetas consisten en una línea que comienza por dos puntos seguidos de uno a ocho caracteres, si se ponen más éstos son ignorados.

IF

Permite una ejecución condicional, de forma que comprueba la condición y ejecuta el comando indicado si ésta se cumple.

Puede utilizarse bajo tres tipos de condición:

— Errores en programas (ERRORLEVEL).

IF ERRORLEVEL 1 ECHO ¡¡¡ESTE PROGRAMA NO FUNCIONA!!!

Si el número de error producido en el programa considerado es 1 saldrá un mensaje en la pantalla.

— Comparar cadenas de caracteres.

IF cadena1==cadena2

Se suelen utilizar con parámetros. Estos parámetros se señalan con el símbolo %, seguido de un número entre 0 y 9, y su utilización es la siguiente:

El comando o programa debe ir seguido de los parámetros correspondientes cuando es tecleado junto al prompt del DOS.

El DOS los identifica por orden con %0, %1,...%9, comenzando por la izquierda, es decir, por el nombre del comando, y terminando por el último parámetro que se haya dado.

Cada vez que en el archivo batch se haga referencia a un %número, se tomará el parámetro correspondiente.

— Comprobar que existe un fichero.

IF EXIST nombre de archivo

o

IF NOT EXIST nombre de archivo

seguidos en cada caso del comando que queremos que se realice en caso de cumplirse la condición.

FOR..IN..DO

Permite la utilización de un comando varias veces. Su sintaxis es:

FOR %%variable IN (nombre/s de archivo/s) DO comando

Los dos signos de %% antes de la variable son importantes para que el DOS no lo confunda con un parámetro.

REM

Permite que aparezca en pantalla un mensaje, siempre y cuando el comando ECHO no haya desconectado el monitor.

SHIFT

Desplaza los parámetros en una lista un lugar hacia la izquierda, borrando el que ocupaba el primer lugar.



UN EJEMPLO DE AUTOEXEC.BAT

Sitúe el disco del DOS en la unidad A, y realice los siguientes pasos:

```
A>COPY CON A:AUTOEXEC.BAT

REM AUTOEXEC EJEMPLO
KEYBSP
DATE
TIME
TYPE %0
B:
DIR
```

pulse F6

aparecerá el mensaje 1 Archivo(s) copiado(s)

Para comprobar su funcionamiento debe poner el sistema en marcha de nuevo. Pulse simultáneamente Ctrl-Alt-Del:

```
A>rem AUTOEXEC EJEMPLO

A>keybsp

A>date
La fecha actual es Vie 24-04-87
Introduzca nueva fecha (dd-mm-aa):
```

```
A>time
La hora actual es 10:06:13,19
Introduzca nueva hora:

A>b:

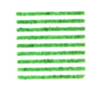
B>dir

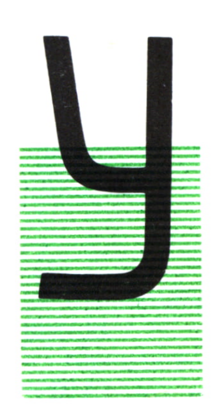
Volumen en unidad B sin etiqueta
Directorio de B:\
LIBRO1     TEX     10     2-4-87
LIBRO2     TEX     12     3-4-87
        2 Archivo(s)      356439 bytes libres

B>
```

DIRECTORIOS Y SUBDIRECTORIOS 5

INTRODUCCION





YA hemos visto que cuando el DOS formatea un disco, crea un directorio con los nombres de los archivos del disco. El número máximo de archivos que caben en ese directorio es:

64 en un diskette de una sola cara.
112 en un diskette de dos caras.
224 en un diskette de alta capacidad.
512 o más en un disco duro.

Para conseguir que el sistema de archivos sea más flexible, el DOS ofrece la posibilidad de crear subdirectorios que dividen al disco en diferentes áreas de almacenamiento, cada una de las cuales se puede utilizar como si fuese un disco diferente.

El directorio principal se conoce con el nombre de directorio raíz y contendrá los nombres de los subdirectorios. Estos, a su vez, podrán tener otros niveles inferiores de almacenamiento, pudiéndose crear una jerarquía completa, denominada sistema de archivos en árbol.



SUBDIRECTORIOS

Un subdirectorio es un archivo que contiene datos como si fuera un directorio, teniendo en cuenta que el formato de salida del mismo es idéntico al del directorio principal que ya conocemos, y que no son válidos los comandos como si se tratara de archivos normales. En los subdirectorios no existen límites de capacidad de archivos como en el directorio principal.

Son prácticamente imprescindibles en el manejo de discos duros, sin embargo, no están restringidos a los mismos, ya que su utilización en diskettes puede facilitar y organizar una labor de archivo.

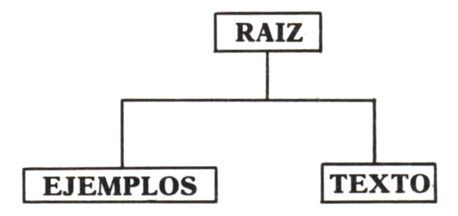
En primer lugar, vamos a crear subdirectorios. Para dar generalidad al capítulo, vamos a realizar los ejemplos sobre un diskette, ya que no siempre se dispone de un disco duro; en cualquier caso, el sistema de utilización de subdirectorios es el mismo con sólo tener en cuenta que habría que referirse a la unidad C (disco duro) en vez de a la A o la B.

Supongamos que deseamos almacenar en un mismo disco el texto correspondiente a los capítulos de un libro escritos con el procesador WordStar y los ejemplos correspondientes al mismo, que por ser un libro sobre hojas de cálculo se han realizado con MultiPlan.

Para una mejor organización de los mismos decidimos la creación de dos subdirectorios:

TEXTO
EJEMPLOS

Estos comienzan a formar nuestra estructura jerárquica:



El comando MKDIR o MD crea un subdirectorio, su sintaxis es:

MKDIR nombre del subdirectorio

Vamos a crear en el disco situado en la unidad B los dos subdirectorios mencionados:

```
B>MKDIR EJEMPLOS

B>MKDIR TEXTO

B>DIR

 Volumen en unidad B sin etiqueta
 Directorio de  B:\

EJEMPLOS      <DIR>      25/04/87  13:38
TEXTO         <DIR>      25/04/87  13:38
        2 Archivo(s)      360448 bytes libres
```

Fig. 1.

En el directorio vemos que los archivos que son subdirectorios aparecen identificados con las letras <DIR> en el lugar que debería ir la extensión. Para referirse al directorio principal o raíz, el DOS utiliza la barra inclinada \ (diferente del signo de división /), en esta ocasión nos está indicando que nos encontramos en el directorio raíz del disco B.

Si ahora le pedimos al DOS que nos muestre uno de los dos subdirectorios tendremos:

```
B>DIR EJEMPLOS

 Volumen en unidad B sin etiqueta
 Directorio de  B:\EJEMPLOS

.            <DIR>      25/04/87  13:38
..           <DIR>      25/04/87  13:38
        2 Archivo(s)     360448 bytes libres
```

Fig. 2.

Evidentemente, no aparecen archivos, puesto que no los hemos introducido.

Nos aparecen a modo de subdirectorios los archivos . y .. seguidos de <DIR>, sin embargo, no se trata de subdirectorios, sino de indicadores que se utilizan para referirse a otros directorios.



CAMBIO DE DIRECTORIO

El DOS mantiene siempre un directorio activo, inicialmente éste será el directorio raíz, pero podemos cambiarlo con el comando CHDIR o CD.

CD nombre del directorio

Cambia el directorio activo por el indicado.

CD

Mostrará cuál es el directorio activo.

```
B>CD
B:\

B>CD TEXTO

B>CD
B:\TEXTO
```

Fig. 3.

El comando DIR y cualquier otro comando que utilicemos del DOS hará referencia al directorio activo.

```
B>DIR

 Volumen en unidad B sin etiqueta
 Directorio de  B:\TEXTO

.             <DIR>       25/04/87  13:38
..            <DIR>       25/04/87  13:38
        2 Archivo(s)       360448 bytes libres
```

Fig. 4.



RUTA DE UN DIRECTORIO

Describe el camino que el DOS sigue a lo largo de los directorios. Supongamos que queremos un archivo llamado HCRENTA.MP que se encuentra en el directorio EJEMPLOS. Su ruta sería:

B:\EJEMPLOS\HCRENTA.MP

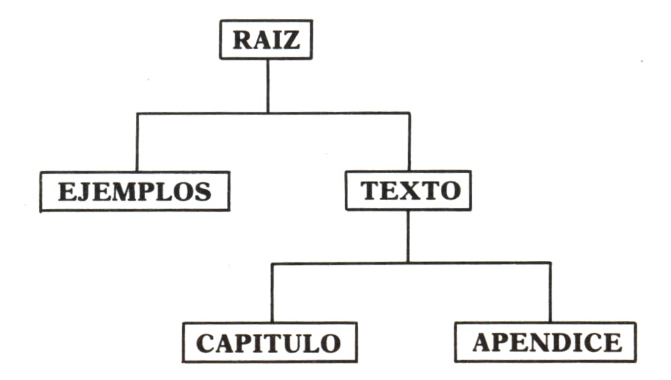
Complicando un poco más el ejemplo, suponemos ahora que en el texto queremos diferenciar entre capítulos y apéndices, para ello creamos una nueva ramificación:

Nos situamos en el directorio de TEXTO:

B>CD TEXTO

En este directorio creamos dos nuevos subdirectorios:

B>MKDIR CAPITULO
B>MKDIR APENDICE



Otra forma de crearlos es desde cualquier directorio, dándole el camino o ruta a seguir:

A>MKDIR \TEXTO\CAPITULO

```
B>MKDIR CAPITULO

B>MKDIR APENDICE

B>DIR

 Volumen en unidad B sin etiqueta
 Directorio de  B:\TEXTO

.            <DIR>      25/04/87  13:38
..           <DIR>      25/04/87  13:38
CAPITULO     <DIR>      25/04/87  13:42
APENDICE     <DIR>      25/04/87  13:42
        4 Archivo(s)     358400 bytes libres
```

Fig. 5.

```
B>DIR \TEXTO\CAPITULO

 Volumen en unidad B sin etiqueta
 Directorio de  B:\TEXTO\CAPITULO

.            <DIR>      25/04/87  13:42
..           <DIR>      25/04/87  13:42
        2 Archivo(s)     358400 bytes libres
```

Fig. 6.

En esta ocasión la ruta de un archivo del subdirectorio APENDICE sería:

\TEXTO\APENDICE\LIBROAP1.LIB

Ya tenemos la estructura del subdirectorio creada y estamos dispuestos para su utilización. Consideraremos cada directorio como si se tratara de discos independientes.

Comenzaremos por copiar un archivo en el directorio RAIZ que como sabemos se representa por la barra inclinada:

B>COPY A:LIBROIND B:\INDICE.LIB

```
B>DIR

 Volumen en unidad B sin etiqueta
 Directorio de  B:\

EJEMPLOS      <DIR>      25/04/87  13:38
TEXTO         <DIR>      25/04/87  13:38
INDICE    LIB      4224   1/01/80   0:38
        3 Archivo(s)     353280 bytes libres
```

Fig. 7.

El siguiente paso será copiar los archivos de ejemplos:

```
B>COPY A:VECINOS.FMD B:\EJEMPLOS
        1 Archivo(s) copiado(s)

B>COPY A:CIRCULAR B:\EJEMPLOS
        1 Archivo(s) copiado(s)
```

Fig. 8.

Y, por último, los archivos de apéndices y capítulos:

```
 Volumen en unidad A sin etiqueta
 Directorio de  A:\

LIBRO5B          11136  15/03/87  18:09
LIBROCPT           896   1/01/80   0:49
LIBROIND          1664   1/01/80   0:04
LIBROAP1          4224   1/01/80   0:38
LIBRO5           32000   8/03/87  13:28
LIBRO12          29440   8/03/87  11:07
LIBRO3           23936   8/03/87  11:42
LIBRO4           41856   8/03/87  13:02
        8 Archivo(s)     214016 bytes libres

B>COPY A:LIBROAP1 B:\TEXTO\APENDICE
        1 Archivo(s) copiado(s)

B>COPY A:LIBRO12  B:\TEXTO\CAPITULO
        1 Archivo(s) copiado(s)
```

```
B>
B>
B>


COPY A:LIBRO3 B:\TEXTO\CAPITULO
        1 Archivo(s) copiado(s)

B>COPY A:LIBRO4 B:\TEXTO\CAPITULO
        1 Archivo(s) copiado(s)
```

Fig. 9.

```
C

B>DIR \EJEMPLOS

 Volumen en unidad B sin etiqueta
 Directorio de  B:\EJEMPLOS

.              <DIR>      25/04/87  13:38
..             <DIR>      25/04/87  13:38
VECINOS  FMD     25600    23/04/87  13:47
CIRCULAR          1280    14/03/87  19:03
        4 Archivo(s)     222208 bytes libres

B>DIR \TEXTO\CAPITULO

 Volumen en unidad B sin etiqueta
 Directorio de  B:\TEXTO\CAPITULO

.              <DIR>      25/04/87  13:42
..             <DIR>      25/04/87  13:42
LIBRO12          29440     8/03/87  11:07
LIBRO3           23936     8/03/87  11:42
LIBRO4           41856     8/03/87  13:02
        5 Archivo(s)     222208 bytes libres

B>DIR \TEXTO\APENDICE

 Volumen en unidad B sin etiqueta
 Directorio de  B:\TEXTO\APENDICE

.              <DIR>      25/04/87  13:42
..             <DIR>      25/04/87  13:42
LIBROAP1          4224     1/01/80   0:38
        3 Archivo(s)     222208 bytes libres
```

Fig. 10.

Observamos un pequeño inconveniente mientras no estamos acostumbrados a trabajar con este tipo de directorios, desconocemos cuál es el directorio activo en cada momento. Hasta ahora la solución observada era utilizar el comando CD o CHDIR sin parámetros que nos daba el directorio activo, pero existe un modo de evitar tener que utilizar el comando cada vez. Cambiamos el prompt del DOS para que nos muestre cada vez el directorio en el que estamos. Esto se hace mediante el comando PROMPT, utilizando el parámetro PG:

B>PROMPT PG

Ahora, el prompt informa de la unidad activa y del directorio.

```
B>PROMPT $P$G

B:\>CD TEXTO

B:\TEXTO>CD CAPITULO
```

Fig. 11.



INDICADORES DEL SUBDIRECTORIO

Vamos a aprender a utilizar esa colección de puntos que aparecen en los diferentes subdirectorios que hemos utilizado.

Los dos puntos (..) representan al directorio inmediatamente superior al que nos hallamos. Podremos, por tanto, referirnos a él con sólo poner los dos puntos, algo que resulta muy útil cuando los nombres de ruta sean largos.

Un ejemplo para aclarar:

B:\TEXTO\CAPITULO>cd ..

B:\TEXTO>

B:\TEXTO>cd ..

B:\>



ULTIMOS COMANDOS

Para poder completar nuestro trabajo con estructuras jerárquicas es útil conocer algunos comandos más:

RMDIR permite suprimir un subdirectorio, su sintaxis es:

RMDIR unidad:ruta

TREE

Este comando no se encuentra en todas las versiones del DOS. Se utiliza para sacar los contenidos de todos los subdirectorios y donde se encuentran. Permite una opción para sacar también el nombre de los archivos. Su sintaxis es la siguiente:

TREE unidad /F

La unidad es el nombre de la unidad de disco que contiene el disco del que deseamos saber la estructura.

/F es opcional, hace que salgan todos los archivos de cada subdirectorio.

```
B:\TEXTO\CAPITULO>TREE B:

LISTADO DE VIAS DE ACCESO AL DIRECTORIO

Vía de acceso: \EJEMPLOS

Sub-directorios:  Ninguno

Vía de acceso: \TEXTO

Sub-directorios:  CAPITULO
                  APENDICE

Vía de acceso: \TEXTO\CAPITULO

Sub-directorios:  Ninguno

Vía de acceso: \TEXTO\APENDICE

Sub-directorios:  Ninguno
```

Fig. 12.

Si su versión del DOS no posee este comando, puede obtener un listado de todos los archivos con extensión <DIR>.

Veamos, como ejemplo final, el listado del contenido o accesos a un disco duro mediante el comando tree:

```
TREE C:

LISTADO DE VIAS DE ACCESO AL DIRECTORIO DEL VOLUMEN
DISCO_DURO

Vía de acceso: \DBASE

Sub-directorios:  Ninguno


Vía de acceso: \TEMPO

Sub-directorios:  Ninguno


Vía de acceso: \AUTOCAD

Sub-directorios:  Ninguno


Vía de acceso: \RUCK

Sub-directorios:  Ninguno


Vía de acceso: \DW3

Sub-directorios:  Ninguno


Vía de acceso: \LX

Sub-directorios:  Ninguno


Vía de acceso: \OA

Sub-directorios:  Ninguno
```

Vía de acceso: \SYMPHONY

Sub-directorios: Ninguno

Vía de acceso: \UTIL

Sub-directorios: Ninguno

Vía de acceso: \123

Sub-directorios: Ninguno

Vía de acceso: \WS

Sub-directorios: Ninguno

Vía de acceso: \SKEY

Sub-directorios: Ninguno

Vía de acceso: \CAT

Sub-directorios: Ninguno

Vía de acceso: \COPIA

Sub-directorios: Ninguno

Vía de acceso: \CONTROL

Sub-directorios: Ninguno

Vía de acceso: \CARRIER

Sub-directorios: Ninguno

```
Vía de acceso: \WORLDS

Sub-directorios:  Ninguno


Vía de acceso: \COPIONES

Sub-directorios:  Ninguno


Vía de acceso: \PROKEY

Sub-directorios:  Ninguno


Vía de acceso: \ACAD

Sub-directorios:  Ninguno


Vía de acceso: \OAII

Sub-directorios:  Ninguno
```

Fig. 13.

O un extracto del mismo poniendo /F (completo resulta excesivamente largo).

Queda por recordar que no pierda de vista las abreviaturas de los comandos vistos en el presente capítulo. Lo normal es trabajar con ellas, por aquello de que el tiempo es oro. Las reproducimos aquí:

MKDIR $\simeq$ MD
CHDIR $\simeq$ CD
RMDIR $\simeq$ RD

Por otra parte, no es conveniente crear en un diskette una estructura tan complicada como la del ejemplo, puede resultar una pérdida de tiempo. Debemos reservar los subdirectorios para casos muy concretos y, sobre todo, para los discos duros.

APLICACIONES DEL DOS 6



INTRODUCCION



EL sistema operativo DOS proporciona el fundamento necesario para la implantación de algunas aplicaciones sencillas. Normalmente, los usuarios utilizan herramientas estándar programadas por diversos fabricantes de la industria. Todas ellas constituyen un complejo mundo de aplicaciones cuyos desarrollos han supuesto gran número de horas y esfuerzo de programación.

Sin embargo, una de las posibilidades que se abren al que se inicia en el entorno de los sistemas operativos es emplear las opciones que éstos facilitan para desarrollar algunas aplicaciones sencillas. A menudo, el DOS sólo supone para el usuario el paso imprescindible hacia el programa que desea emplear. Este apéndice recoge algunas posibilidades cuya utilización puede proporcionarnos un interesante conjunto de nuevas opciones.



EN LA INICIALIZACION

Como ya sabemos, siempre que se arranca el ordenador, el DOS trata de localizar un fichero cuyo nombre es AUTOEXEC.BAT. Este fichero está formado por una lista de comandos que queremos que se ejecuten sin necesidad de teclearlos cada vez al iniciar el sistema. Estos comandos pueden emplearse tanto para inicializar parámetros del sistema, arrancar un programa de forma automática o visualizar en la pantalla mensajes o comentarios.

El fichero AUTOEXEC.BAT que se muestra a continuación es de gran utilidad y puede ser la base de otros más complejos:

```
PROMPT $p$g
PATH a:\;b:\;b:\DOS
DATE
REM Bienvenido al sistema
```

El primer comando convierte el símbolo de indicación de mensaje:

A>

en un indicador del directorio en que nos encontramos trabajando. Este dato resulta de especial trascendencia para aquellos que se encuentren trabajando con disco fijo, ya que en un disco flexible la utilización de directorios es mucho menor. Así, es posible conocer no sólo la unidad en que se opera, sino también el directorio. Con el segundo comando se señala al sistema el camino de búsqueda de ficheros. Con él tratará en primer lugar de localizar el fichero en la unidad en que esté y si no lo encuentra pasará a buscarlo a las siguientes unidades o directorios señalados en el comando PATH.

Con ello se consigue evitar la tediosa tarea de tener los programas en una unidad en concreto, basta un intento de localización y el sistema empleará el camino marcado para realizar la búsqueda. Es posible, además, incluir en el camino de búsqueda diferentes directorios. Se consigue así un ahorro en la ocupación de disco fijo, puesto que no es necesario repetir una copia de algunos programas en todos los directorios.

En el ejemplo, al intentar realizar la búsqueda de un fichero en la unidad a y no encontrarlo, pasaría a buscar en la unidad b y, posteriormente, en el directorio DOS de dicha unidad. Supongamos que estamos trabajando con una determinada hoja de cálculo y vamos a almacenar los datos en un disco flexible que es necesario formatear. Con la asignación de directorios y caminos anterior podríamos llamar al comando FORMAT (en el directorio b:\DOS) sin necesidad de abandonar el directorio de la hoja de cálculo en que estemos.

El comando DATE solicita la fecha. La pricipal utilidad de este comando viene dada por la forma de funcionar del DOS. Este, cuando escribe un fichero en disco, anota la fecha de su última modificación y así es posible conocer el estado y situación en el tiempo de nuestros ficheros, programas, etc.



INICIO DE APLICACIONES

Supongamos que tratamos de construir un fichero autoexec.bat para que, a partir de él, se efectúe el arranque de una aplicación en concreto. Consideramos una aplicación cuyo comando de arranque es TEXTO. Bastaría añadir al fichero AUTOEXEC.BAT anterior dos líneas como:

```
REM Arranque del programa TEXTO
TEXTO
```

Con ello, después de tomar los datos ya explicados, se arrancaría la aplicación cuyo comando de inicio es TEXTO.



SISTEMAS DE MENUS

El uso de menús en el trabajo resulta de gran utilidad tanto para los usuarios expertos como para los más noveles. En este apartado vamos a sugerir algunos de los posibles diseños de un menú empleando las posibilidades del sistema operativo y en particular de los ficheros BATCH (.BAT).



USANDO ECHO

En el primer método se incluye un menú en el fichero AUTOEXEC.BAT. El comando ECHO se emplea para visualizar cada línea. Una forma de probarlo es generar un fichero que se llame EXEC.BAT y ejecutarlo simplemente tecleando EXEC (no es necesario volver a arrancar el sistema para ejecutar un fichero BATCH). Por ejemplo, podríamos usar un fichero como este:

```
ECHO OFF
CLS
PROMPT $p$g
PATH A:\;B:\
DATE
ECHO BIENVENIDO AL SISTEMA
ECHO .....................
ECHO LAS APLICACIONES SIGUIENTES
ECHO ESTAN DISPONIBLES
ECHO ................
ECHO Wordstar (teclee WS)
ECHO dBase III (teclee DBASE)
```

Con este menú se orienta al que se conecta sobre las aplicaciones de que dispone y cómo iniciar su uso



EMPLEO DEL COMANDO TYPE

Cuando se emplea el comando ECHO se obtiene un rendimiento bastante penoso, puesto que en cada línea se procesa el comando retardando mucho su ejecución. Para conseguir una mejor presentación y más rápida es, sin duda, mejor emplear el comando TYPE. Este comando visualiza un fichero sobre la pantalla. En este segundo método empleamos tres ficheros denominados:

AUTOEXEC.BAT

MENU.BAT
MENU.TXT

El fichero MENU.TXT contiene todo el texto del menú que deseamos que aparezca en pantalla. Dentro del fichero AUTOEXEC.BAT se ejecuta el comando TYPE que lo presenta sobre la pantalla. Además se genera el fichero MENU.BAT que permite volver al menú desde el DOS tecleando simplemente MENU.

En primer lugar se crea un fichero con nombre MENU.TXT y cuyo contenido puede ser:

```
Bien venido al sistema.
Las aplicaciones siguientes
se hallan disponibles.


Wordstar (teclee WS)
dBASE III (teclee DBASE)
```

Este fichero puede crearse tanto con un procesador de texto como con el editor del DOS (EDLIN).

A continuación habrá que modificar el fichero AUTOEXEC.BAT para que visualice el menú. Para ello basta incluir la siguiente línea:

```
TYPE MENU.TXT
```

Por último, será conveniente disponer de una posibilidad de acceder al menú general sin necesidad de pasar por el AUTOEXEC.BAT. Por eso se crea el fichero MENU.BAT. Su contenido podría ser el siguiente:

```
ECHO OFF
CLS
TYPE MENU.TXT
```

Y así cada vez que se teclee MENU en el DOS se obtendrá el menú en pantalla. Por supuesto, que este ejemplo sólo trata de mostrar las posibilidades de que se dispone en el DOS. Es evidente que para el caso de tener dos aplicaciones únicamente puede no resultar cómodo un juego de menús. Sin embargo, sí es interesante procurar aplicar estos conceptos siempre que nos sea posible, ya que facilitan enormemente el trabajo.



MENUS AUTOMATICOS

Es posible incrementar aún más la productividad del sistema de menús añadiendo números a éste. Para ello identificaremos cada aplicación con una cifra. En el ejemplo que se muestra a continuación cada opción posible va precedida por un número. Al introducir este número se ejecuta

el fichero BATCH correspondiente que arranca el programa seleccionado. El fichero MENU.TXT podría ser:

```
SELECCIONE LA OPCION Y PULSE <INTRO>
1. WORDSTAR
2. dBASE III
3. MULTIPLAN
4. Vuelta al DOS
```

Y el fichero autoexec.bat sería el mismo que se analizó en el caso anterior. Quedan por describir los ficheros que entrarían a ejecutarse al seleccionar cada opción y que dan paso a cada uno de los programas. Estos serían:

Fichero.................1.BAT

```
ECHO OFF
CLS
WS
CLS
TYPE MENU.TXT
```

Fichero.................2.BAT

```
ECHO OFF
CLS
DBASE
CLS
TYPE MENU.TXT
```

Fichero.................3.BAT

```
ECHO OFF
CLS
MP80
CLS
TYPE MENU.TXT
```

Fichero.................4.BAT

```
ECHO OFF
CLS
```

Con ello se consigue un sistema de menús completo, puesto que cada opción del menú selecciona un programa, una vez finalizada la utilización del programa se vuelve a visualizar el menú principal (MENU.TXT). En el

caso de la opción 4 basta no volver a «llamar al menú» para retornar directamente al sistema. Este tipo de disposición de menús permite aprovechar el DOS para incrementar la productividad en el manejo de aplicaciones.



OTROS TIPOS DE MENU

La gran variedad de menús posibles a crear deja abierto un extenso campo de posibilidades. Sin embargo, existen algunos casos de especial relevancia y que conviene reseñar por su interés general. A menudo, la iniciación en el manejo del DOS se hace incómodo y se echa en falta disponer de una ayuda en pantalla como la que existe en multitud de aplicaciones. En el DOS es necesario recurrir al manual para resolver dudas sencillas. Para estos casos es francamente útil concebir un conjunto de ayudas o de ficheros con información sobre la sintaxis de algunos comandos del DOS.

Una posible opción consiste en construir un menú llamado AYUDA.TXT que se utilice a partir de un fichero llamado AYUDA.BAT como el siguiente:

```
ECHO OFF
CLS
TYPE AYUDA.TXT
```

El fichero AYUDA.TXT puede contener cualquier tipo de información que consideremos relevante o de verdadera utilidad como ayuda de trabajo. En caso de que ocupara más de una pantalla se puede incluir el comando MORE del DOS para generar una pausa después de la primera visualización. Con ello, el fichero AYUDA.BAT quedaría como sigue:

```
ECHO OFF
CLS
MORE < AYUDA.TXT
```

Se podría también llamar a otros menús desde el principal para acceder a otros ficheros de ayuda.



MENUS EN DIRECTORIOS

Los usuarios que dispongan de una unidad de disco fijo pueden aprovechar aún más esta configuración a base de un menú principal y varios secundarios. Supuesto que cada aplicación estuviera en un directorio, podría generarse un conjunto de menús independientes unos de otros. Por ejemplo, podría hacerse que al seleccionar la opción 1 Wordstar del menú

principal se pasara a un directorio exclusivo de la aplicación. El fichero 1.BAT correspondiente quedaría:

```
ECHO OFF
CLS
CD WORDSTAR
TYPE MENU.TXT
```

con lo que llamaría a un fichero llamado MENU.TXT, pero que sería distinto del MENU.TXT de la aplicación dBASE III. Una vez finalizado el trabajo en el directorio correspondiente sería necesario volver al menú principal, con lo que se podría incorporar una opción en el MENU.TXT del directorio que fuera «RETORNO AL MENU PRINCIPAL». En ella se incluiría el comando:

```
CD \
```

de vuelta al directorio principal en vez del de paso al de la aplicación.

El hecho de la facilidad de uso de los ficheros BATCH hace de ellos una herramienta práctica y de uso sencillo en el entorno del DOS. Los sistemas de menús explicados pueden integrarse de forma sencilla en una configuración de disco fijo. Una posibilidad consiste en crear un directorio para los ficheros BAT y que los ficheros MENU.TXT se hallen en los directorios correspondientes.

Con ello sería posible ahorrarse la duplicidad de algunos ficheros como el MENU.BAT o el AYUDA.BAT. Debe tenerse en cuenta que la utilización de ficheros pequeños desperdicia enormemente el espacio en disco. Por ejemplo, el mínimo espacio de asignación en un PC-XT es de 4 Kbytes.



MAS DIFICIL TODAVIA

Una interesante opción para mejorar un poco el manejo de menús consiste en incluir el funcionamiento de las teclas de función. Para realizar la asignación de teclas debe cargarse en primer lugar el fichero ANSI.SYS. Ello se efectúa incluyendo el comando:

```
DEVICE=ANSI.SYS
```

en un fichero cuyo nombre es CONFIG.SYS. Para asignar la opción de AYUDA a la tecla de función F9 y la opción de vuelta al menú a la tecla F10 deben incluirse los siguientes comandos en el fichero AUTOEXEC.BAT:

```
PROMPT $e[0;67;"AYUDA";13p
PROMPT $e[0;68;"MENU";13P
```

Una vez se han efectuado las indicaciones anteriores en los ficheros

CONFIG.SYS y AUTOEXEC.BAT es necesario volver a arrancar el sistema para conseguir la inicialización de las teclas. Según lo anterior basta pulsar F9 y accederemos al programa de ayuda.

El lector se habrá dado cuenta de las inmensas posibilidades que abre esta nueva opción del DOS. Con ello podemos disponer de un juego de teclas de función para una determinada aplicación y configurar un nuevo conjunto para otro tipo de aplicación. Será también interesante tener un juego personal de teclas de función para cada usuario del PC, ya que sus necesidades pueden ser distintas.

NUEVAS POSIBILIDADES DE LOS FICHEROS POR LOTES

Los ficheros por lotes descubren o facilitan muchas más posibilidades de las que hemos citado. No sólo es posible realizar menús o arrancar aplicaciones. Los ejemplos que figuran a continuación muestran formas de ampliar esas funciones y automatizar el trabajo. Una vez comprendida la forma de llevar a cabo este tipo de aplicación nuestra imaginación o nuestro tiempo serán los límites de lo que podamos crear.



FICHEROS REPETITIVOS

Las versiones 2 y 3 del DOS incluyen el subcomando GOTO dentro de los ficheros BATCH. Este puede utilizarse para realizar tareas de forma repetitiva hasta que el usuario decida detenerlo. El fichero COPIAR.BAT puede usarse para realizar la copia de un conjunto de ficheros a una serie de discos sin tener que volver a teclear cada vez los nombres.

```
:LOOP
REM Sitúe el nuevo disco en la unidad B
PAUSE
COPY A:NOMFICH1.TXT B:
COPY A:NOMFICH2.TXT B:
COPY A:NOMFICH3.TXT B:
GOTO LOOP
```

La primera línea del fichero es una etiqueta (identificada como tal por los puntos que preceden al nombre). Esta etiqueta se emplea en la última línea para provocar la repetición del conjunto de comandos comprendido entre ambas. (Para detener la ejecución de este fichero es necesario pulsar simultáneamente las teclas CONTROL y BREAK.) El PAUSE permite cambiar de disco para continuar la copia de los programas. Una vez que

se ejecutan todos los COPY se puede volvar al comienzo con el comando GOTO.

Otra posible aplicación de los ficheros repetitivos puede consistir en la creación e impresión de direcciones de remite para envíos postales. Se crea un fichero IMPRIME.BAT que realiza la impresión del remite que se halla en un fichero REMITE.TXT sobre una etiqueta o un sobre, repitiendo el proceso hasta finalizar las direcciones que sean necesarias. Cuando se hallan impreso suficientes, el usuario detendrá la impresión. El fichero, supuesta una impresión continua sin paradas, sería:

```
:BUCLE
COPY REMITE.TXT LPT1:
GOTO BUCLE
```

Un ejemplo del fichero REMITE.TXT como el que se muestra a continuación puede ahorrarnos trabajo.

SOLEDAD TAMARIZ-MARTEL MORENO
C/ General López Pozas, 12
Madrid-28036

Esta misma posibilidad de impresión puede valer para imprimir juego de cartas o impresos o cualquier tipo de fichero que nos sea de utilidad.

La potencia del DOS puede ser aumentada empleando las propias opciones del DOS para automatizar muchas de sus funciones. Por ejemplo, si se emplean subdirectorios con frecuencia puede habilitarse un comando que evite tener que teclear cada vez para acceder a un subdirectorio determinado. Por ejemplo, para tener acceso a un cuarto nivel de directorio bastará teclear FEB si hemos creado un fichero FEB.BAT, cuyo contenido sea:

```
CD \MIDIR\WS\CARTAS\FEB
```

El siguiente fichero evita tener que teclear un comando del DOS que se usa con relativa frecuencia. El fichero funciona como una macro de teclado. En el fichero DIRFECHA.BAT se clasifica un directorio por fecha (columna 33) y se envía a la impresora utilizando las opciones de redireccionamiento de entrada/salida.

```
DIR 1 SORT /+33 >LPT1
```



MEJORAS EN LA IMPRESION

A menudo es difícil recordar el número del código que identifica un tipo de carácter de impresión. Es más, en ocasiones es necesario buscar en el manual de la impresora para recordar el carácter que obtenía letra

más pequeña o enfatizada. Bien, para facilitar esta labor podemos elaborar un conjunto de ficheros de impresión. Cada uno de ellos dispondrá de un nombre sencillo de recordar que facilitará la labor de impresión sin tener que estar recordando cadenas de caracteres. Por ejemplo, en una impresora matricial IBM para obtener enfatizado en la impresión sería necesario un fichero como el que se muestra a continuación.

```
ECHO (Alt-155)E >lpt1
```

cuyo nombre podría ser ENFATIZA.BAT. El carácter ESCAPE (aquí representado por Alt-155) se obtiene manteniendo pulsada la tecla ALT y tecleando 155 en la zona de teclado numérico. Al soltar la tecla ALT se visualiza el carácter.

Empleando una técnica similar se puede construir todo un conjunto de ficheros para tener preparada la impresión de ficheros en diferentes tipos de letras. Los códigos que identifican cada tipo dependen de la impresora en concreto y están en el manual de cada impresora.

El ejemplo que se muestra a continuación imprime un fichero en formato comprimido y después deja la impresora en su estado normal. Además, este fichero hace uso de la opción de variables del DOS. Tecleando el nombre del fichero BAT y a continuación el nombre del fichero sobre el que actúa el comando se da la orden. Este fichero, sobre el que provoca la acción, está representado en el fichero BAT mediante el símbolo %1, ya que es la primera variable. Si se emplearan más, las sucesivas referencias serían %2, %3 etc.

El fichero sería el siguiente:

```
ECHO (Alt-15) > lpt1:
COPY %1 LPT1:
ECHO (Alt-155)@> lpt1:
```

El nombre podría ser COMPRIME.BAT y, por tanto, para aplicarlo sobre el fichero PRUEBA.TXT teclearíamos COMPRIME PRUEBA.TXT. Así se sustituiría la variable %1 por el fichero de texto que deseamos imprimir. (Más adelante volveremos a tratar este tipo de parámetros.) La última línea devuelve la impresora a su estado inicial.

A medida que se van creando nuevas utilidades es posible que se quiera generar menús que las vayan describiendo. Este menú podría llamarse UTIL.TXT y podría visualizarse en pantalla empleando un fichero UTIL.BAT similar al utilizado en el caso de los AYUDA.BAT y MENU.BAT ya descritos.



REDIRECCIONAMIENTO DE LA ENTRADA/SALIDA

La capacidad de redireccionar información de E/S desde y hacia fuentes o destinos se añadió al DOS en marzo de 1983 en la versión 2.0. Esta posibilidad ya la hemos incluido en algunos de los ejemplos que hemos mostrado anteriormente, pero no por ello dejaremos de explicar sus enormes posibilidades de aplicación y la extensa variedad de usos a los que aplicarla.

Por ejemplo, usando redireccionamiento es posible enviar un texto que de forma estándar aparecería en pantalla un fichero o una impresora con un simple tecleo del comando correspondiente. De la misma forma, es posible direccionar texto a un programa o a un comando del DOS. Una técnica ampliamente utilizada y ya clásica consiste en emplear un fichero para responder preguntas realizadas por diferentes comandos del DOS como el FORMAT o el DISKCOPY. Para ilustrar la cuestión con un ejemplo, consideremos un fichero denominado: NO.ANS:

<RETURN>
N

Es posible utilizar el redireccionamiento con este sencillo fichero de la siguiente forma. Teclee:

A>DISKCOPY A: B: >NO

Cuando se ejecute este comando, el carácter de retorno de carro de la primera línea del fichero NO.ANS se emplea en el momento en que el comando DISKCOPY solicita del usuario que pulse cualquier tecla para continuar. La N se utiliza cuando el DISKCOPY pregunta si se desea realizar alguna otra copia. Empleando este procedimiento no es necesario sentarse frente al teclado durante toda la operación.

GRABACION DE LA HORA A EFECTOS DE CONTABILIDAD

También es posible aplicar el direccionamiento para mantener un registro del tiempo que se está utilizando el sistema. Para ello se sitúan en el fichero AUTOEXEC.BAT las líneas siguientes:

DATE >> HORAGRAB.DAT
ECHO HORA DE CONEXION: >> HORAGRAB.DAT
TIME >> HORAGRAB.DAT

Además será necesario disponer de un fichero SALIR.BAT que efectúe las operaciones de despedida del sistema. Estas serán sus líneas:

```
ECHO HORA DE CIERRE: >> HORAGRAB.DAT
TIME >> HORAGRAB.DAT
ECHO ——>> HORAGRAB.DAT
```

Si se inicializa el sistema con estos dos ficheros así preparados (las líneas del AUTOEXEC.BAT y el SALIR.BAT), el DOS pedirá la fecha y la hora de conexión y a continuación creará un fichero de nombre HORAGRAB.DAT en el que se grabará la fecha, el mensaje «HORA DE CONEXION» y la hora.

Observe que se utiliza el signo > de forma doble (>>). Si se hubiera efectuado la misma operación con el símbolo simple, cada vez que se abriera el fichero HORAGRAB.DAT los datos a introducir se teclearían sobre los ya existentes, de esta forma se perdería información cada vez que se efectuara la conexión al ordenador. Sin embargo, el símbolo doble >> indica que los nuevos datos a introducir se sitúen después de los últimos valores introducidos. Así se consigue un fichero de registro de horarios muy completo. Por supuesto, este método requiere que el usuario recuerde que debe arrancar el SALIR.BAT cada vez que abandone el sistema. Este fichero realiza las operaciones inversas al anterior. Graba la hora y traza una línea que permita diferenciar con facilidad los distintos momentos de conexión al ordenador a lo largo del tiempo. Para ver y analizar el registro horario bastará teclear:

```
TYPE HORAGRAB.DAT
```

o reenviarlo a una impresora con los métodos ya descritos.



UTILIZACION DE PARAMETROS SUSTITUIBLES

Se han hecho algunos comentarios acerca de la utilización de los parámetros reemplazables o variables. Sin embargo, conviene ahondar sobre su utilización. Un parámetro sustituible es una variable en un fichero BAT que se sustituye directamente por la cadena de caracteres que se teclea en la línea del comando para ejecutar el fichero BAT.

Los nombres de variable empiezan por el símbolo % y un número del 0 al 9 a continuación (o superior si se emplea el subcomando SHIFT para ficheros por lotes). Al ejecutar un fichero BAT, el propio nombre del fichero se sitúa en la variable %0. Por tanto, la primera variable que es posible utilizar es %1, que será la primera cadena tecleada a continuación del nombre del fichero BAT; la segunda será %2, y así sucesivamente. A partir de este momento nos referiremos a estas variables como los parámetros.

La razón fundamental para el empleo de los parámetros en ficheros BAT es poder hacerlos más funcionales y poderlos aplicar en un mayor número de campos. De esta forma los ficheros por lotes pasan a ser más útiles, ya que es posible especificar valores diferentes para los parámetros cada vez que se ejecutan.

Recordemos el fichero COMPRIME.BAT que imprimía el fichero que se especificaba en la línea del comando en forma comprimida. En él la primera línea situaba la impresora en modo comprimido y la última lo liberaba. Este fichero puede utilizarse para imprimir en formato comprimido cualquier fichero, con tal de especificar su nombre en la línea de comando a continuación del nombre del fichero por lotes (COMPRIME). Si no se indica ningún nombre de fichero, no se enviará nada a la impresora, puesto que la variable %1 no contiene nada. A veces puede resultar interesante disponer de técnicas que aseguren que se han tecleado los ficheros necesarios para cubrir las variables que incluye un fichero de este tipo. Para ello explicamos el ejemplo que figura a continuación.

El fichero que se muestra es el COMPRIME.BAT, pero modificado. En él se incluye un sencillo procedimiento de verificación de que se ha dado el nombre del fichero a imprimir, y en caso contrario solicita del usuario el nombre del fichero a imprimir:

```
ECHO OFF
CLS
IF “%1“ == “ “ GOTO NOFICH
ECHO (Alt-15) > LPT1:
COPY %1 LPT1:
ECHO (Alt-155)@> LPT1:
GOTO FINAL
:NOFICH
ECHO Nombre del fichero a imprimir?
:FINAL
```

Si el parámetro %1 no tiene contenido, en la tercera línea se produce un salto a la etiqueta NOFICH, se visualiza el mensaje: Nombre de fichero a imprimir? y finaliza el proceso. En el caso de haber tecleado un nombre de fichero, éste se envía a la impresora y se efectúa un salto a la etiqueta FINAL. Este procedimiento emplea el subcomando de ficheros por lotes IF que compara las cadenas entre comillas. Si no se introduce ningún nombre de fichero, no habrá nada entre las comillas del %1, lo que hace que sea cierto lo que señala la sentencia IF. De ser este el resultado de la comparación se produce el salto a NOFICH.

Cuando un fichero de este estilo va a ser utilizado por principiantes, suele ser más efectivo y menos «fuerte» visualizar un mensaje del aspecto siguiente:

```
ECHO FICHERO PARA IMPRIMIR COMPRIMIDO
ECHO Para usar correctamente este
ECHO este fichero debe introducirse
ECHO el comando de la siguiente forma
ECHO
ECHO COMPRIMIR Nombre de fichero
ECHO ***********************************
ECHO Para imprimir el fichero TEXTO.TXT
ECHO sería: COMPRIMIR TEXTO.TXT
ECHO ***********************************
```



SELECCION DE OPCIONES MEDIANTE FICHEROS BAT

A medida que se va adquiriendo práctica en la escritura y diseño de ficheros por lotes, y sus comandos correspondientes se descubre como una interesante posibilidad la elaboración de ficheros que permite la selección de opciones dentro de ellos. En el ejemplo que se da a continuación, se muestra cómo visualizar el listado de un directorio con diferentes formatos dependiendo de los caracteres introducidos como comando.

Si se teclea ORDENPOR NOMBRE el directorio del disco aparecerá con los ficheros ordenados alfabéticamente. Tecleando ORDENPOR FECHA ordena la lista por la fecha de cada archivo. Además es posible clasificar por tamaño y por extensión. Al teclear ORDENPOR se visualiza una pequeña ayuda del procedimiento.

```
ECHO OFF
CLS
IF "%1" == "" GOTO AYUDA
GOTO %1
:AYUDA
ECHO Para utilizar este programa
ECHO teclee el nombre del fichero
ECHO por lotes (ORDENPOR) seguido
ECHO de una de las siguientes opciones
ECHO NOMBRE....para ordenar alfabéticamente.
ECHO FECHA.....para ordenar por fecha.
ECHO TAMA......para ordenar por tamaño.
ECHO EXT.......para ordenar por extensión.
GOTO FINAL
:NOMBRE
DIR ˙SORT
GOTO FINAL
```

```
:FECHA
DIR 'SORT /+24
GOTO FINAL
:TAMA
DIR 'SORT /+17
GOTO FINAL
:EXT
DIR 'SORT /+10
GOTO FINAL
:FINAL
```

La línea 4 del fichero es la que produce las bifurcaciones a las distintas etiquetas reseñadas. De esta forma se obtiene el listado del directorio de la forma solicitada y se visualiza en la pantalla. Se podría enviar cada uno de estos listados a la impresora o a un fichero simplemente añadiendo los comandos ya mencionados de redireccionamiento. (>PRN o LPT1 y/o >TEMPORAL.TXT) al final de cada comando DIR.



UTILIDADES AVANZADAS

Los dos ficheros que figuran a continuación recogen la mayor parte de las aplicaciones que hemos estado comentando a lo largo de estas líneas. El primero de ellos, denominado COLOR.BAT, permite alterar fácilmente el color de un monitor gráfico de color. Antes de ejecutarlo por primera vez debe añadirse la línea siguiente:

```
DEVICE = ANSI.SYS
```

en el fichero CONFIG.SYS y rearrancar el sistema. El fichero queda como sigue:

```
ECHO OFF
CLS
IF “%1“ == ““ GOTO AYUDA
GOTO %1
:AYUDA
ECHO Este fichero permite cambiar
ECHO los colores de su monitor.
ECHO Para cambiarlos teclee COLOR
ECHO seguido del color què desea
ECHO ROJO, VERDE, AZUL o RESET
GOTO FINAL
:ROJO
PROMPT $e[31m
GOTO FINAL
```

```
:VERDE
PROMPT $e[32m
GOTO FINAL
:AZUL
PROMPT $e[34m
GOTO FINAL
:RESET
PROMPT $e[0m
GOTO FINAL
:FINAL
PROMPT $p$g
```

Pueden incluirse incluso más colores si se desea y el monitor lo permite. Los ficheros BAT de gran tamaño se ejecutan con mayor rapidez sobre memoria RAM; por ello, si se dispone de un disco virtual en memoria resulta recomendable copiarlos sobre él.

El siguiente fichero por lotes permite cambiar e incluso combinar diferentes tipos de letras de su impresora IBM o compatible. Es posible incluso realizar más de una selección de forma simultánea. En el ejemplo se muestra la utilización del subcomando SHIFT para aumentar el número de parámetros por si se pretende disponer de más de un estilo de letra. Al teclear el fichero recuerde el tratamiento que reciben los caracteres consignados como (Alt-número). El fichero se llamará PONTIPO.BAT.

```
ECHO OFF
CLS
IF "%1" == "" GOTO AYUDA
ECHO ELECCION DE TIPOS DE LETRA
GOTO %1
:BUCLE
SHIFT
IF "%1" == "" GOTO FINAL
GOTO %1
:AYUDA
ECHO UTILICE LOS CODIGOS QUE SE MUESTRAN
ECHO COMO PARAMETROS PARA EL FICHERO PONTIPO.BAT
ECHO ====================================
ECHO EXPAN (Activa letra expandida)
ECHO COND (Activa letra condensada)
ECHO SUBR (Activa el subrayado)
ECHO ENFAT (Activa el enfatizado)
ECHO DOBLE (Activa la doble pasada)
ECHO RESET (Devuelve al modo normal)
GOTO FINAL
:EXPAN
```

```
ECHO (Alt-155)W1 > LPT1
GOTO BUCLE
:COND
ECHO (Alt-15) > LPT1
GOTO BUCLE
:SUBR
ECHO (Alt-155)-1 >LPT1
GOTO BUCLE
:ENFAT
ECHO (Alt-155)E >LPT1
GOTO BUCLE
:DOBLE
ECHO (Alt-155)G >LPT1
GOTO BUCLE
:RESET
ECHO (Alt-155)@>LPT1
:FINAL
```

El fichero **PONTIPO.BAT** dispone de interesantes opciones que pueden resultar de utilidad en otros ficheros por lotes. Pueden especificar varios parámetros en la línea de comandos. Por ejemplo, si se teclea **PONTIPO EXPAN DOBLE**, el fichero **BAT** tomará el parámetro **EXPAN** como primer parámetro (%1) saltará a :EXPAN y preparará la impresora con letra expandida. Después de activar este tipo, se producirá un salto volviendo a :BUCLE y los parámetros se considerarán desplazamientos sobre el primero. Así, se tomará **DOBLE** como primer parámetro repitiéndose la secuencia anterior. Se activa la opción en la impresora y se salta a :BUCLE al igual que antes. La tercera vez, sin embargo, el comando **IF** detecta una variable en blanco, lo que produce un salto al final del fichero **PONTIPO.BAT**.



SISTEMAS CON PALABRA DE PASO

La mayor parte de las oficinas o centros de trabajo disponen de usuarios de PC que acceden a los mismos programas y datos. Si bien deben habilitarse medios potentes de protección de los datos más confidenciales o más críticos de la empresa, a veces puede ser suficiente con un sistema sencillo de «palabra de paso» (password). La primera rutina que se muestra a continuación es la más sencilla. Incluye el uso del fichero **AUTOEXEC.BAT** y de un fichero especial denominado **LOGON.BAT**. El fichero **AUTOEXEC.BAT** únicamente borra la pantalla y visualiza el mensaje:

```
ECHO OFF
CLS
```

```
ECHO Teclee LOGON, un espacio
ECHO y a continuacion su clave.
```

Supuesto que su palabra clave fuera LUIS teclearía LOGON LUIS. El fichero LOGON.BAT que se ejecutaría a continuación sería:

```
ECHO OFF
CLS
IF "%1" == "" GOTO REINTENTE
IF "%1" == "QUECA" GOTO OK
IF "%1" == "LUIS" GOTO OK
ECHO ACCESO DENEGADO
GOTO FINAL
:REINTENTE
ECHO PALABRA CLAVE??
GOTO FINAL
:OK
TYPE MENU.TXT
:FINAL
```

El fichero LOGON.BAT compara la palabra clave introducida en la línea de comando con las indicadas en las líneas 4 y 5. Si coincide con alguna de ellas se permite el acceso y si no se deniega.

Es posible generar sistemas más completos de protección. Por ejemplo, puede hacerse que no aparezcan en pantalla los caracteres tecleados. Para ello basta incluir en el AUTOEXEC.BAT el comando:

```
PROMPT $e[8m
```

Podrían, además, complicarse los procedimientos de acceso con la introducción de más variables de acceso, evitando la posibilidad de utilizar la tecla BREAK, etc.



RESUMEN

Este capítulo sólo ha tratado de ser una mera aproximación a algunas de las posibilidades que ofrece un sistema operativo como el DOS. A veces, se supone que un sistema es únicamente el soporte básico para superponer y ejecutar aplicaciones más complejas. Hemos tratado de mostrar que también las posibilidades del DOS proporcionan un amplio juego de opciones para generar utilidades que incrementan nuestra productividad y comodidad al trabajar.

APENDICE 1



ULTIMA VERSION DEL DOS

La versión 3.3 del DOS de IBM es la más reciente de este sistema operativo de una sola tarea para la familia del ordenador personal de IBM y del IBM Personal System/2, el nuevo ordenador personal de IBM presentado en España en abril del 87.

Esta versión es totalmente compatible con la versión 3.2 y soporta una amplia gama de productos de comunicaciones y aplicaciones. El manual «Technical Reference» proporciona información técnica y programas auxiliares adicionales.

Ofrece nuevas prestaciones, entre las que cabe citar el comando APPEND (acceso a archivos externos al directorio actual); el comando ATTRIB, que permite la modificación de la información reflejada en el directorio de archivos, y mejoras como las efectuadas en el soporte de la conexión serie, en la ejecución de archivos por lotes y en el rendimiento general.

APENDICE 2



PRINCIPALES COMANDOS DEL MS-DOS



Introducción

Presentamos aquí una visión de los comandos que posee el DOS con una breve explicación de la función que realizan. No deje pasar por alto este apéndice como si sólo se tratase de una amplia lista de comandos, léala, no se la estudie, no es necesario, sin embargo, comprobará que el saber que existe un comando para una determinada función le permite acudir a su guía del DOS con alguna ventaja.



CLASIFICACION

Los comandos del DOS se clasifican en dos tipos:

— INTERNOS
— EXTERNOS

Los comandos internos se ejecutan inmediatamente porque están incluidos dentro del DOS, por tanto, no es necesario que el diskette del DOS permanezca insertado en la unidad para ejecutar estos mandatos.

Los comandos externos residen en el disco como archivos de programa; por tanto, deben leerse en el disco antes de ejecutarse.

Cualquier archivo con la extensión .BAT, .COM o .EXE se considera un mandato externo. Esto permite que cualquiera programe sus propios mandatos y los añada al sistema. Por otra parte, cuando se introduce una orden no es necesario añadir su extensión.



COMANDOS POR ORDEN ALFABETICO

ASSIGN

Tipo externo

Indica al DOS que transforme las peticiones de entrada/salida para una unidad, en peticiones de entrada/salida para otra unidad. Es muy útil para trasladar la llamada a una unidad de disco (la A) a otra unidad de disco (la C, por ejemplo).

Su sintaxis es:

ASSIGN <unidad 1>=<unidad 2>

Si se omiten los parámetros el efecto es que el DOS anula todas las asignaciones vigentes hasta entonces.

ATTRIB

Tipo externo

Permite definir el atributo de un archivo como de sólo lectura o visualizar el atributo de un archivo.

Su sintaxis es:

ATTRIB+R<nombre archivo>

Para poner atributo de sólo lectura.

ATRRIB-R<nombre de archivo>

Para permitir modificaciones o el borrado del archivo.

ATTRIB <nombre del archivo>

Indicará con una R si el archivo es de sólo lectura.

Permite la utilización de caracteres comodín para modificar una serie de archivos.

BACKUP

Tipo externo

Realiza copias de seguridad de uno o más archivos de disco a otro disco. La especificación de cada unidad de disco debe ser diferente.

Su sintaxis es:

BACKUP <ruta><nombre de archivo> <unidad> /A /S /M /D:<fecha>

<ruta> es el nombre de la ruta del directorio donde se encuentran los archivos que se van a copiar. Si no se especifica ninguno, el DOS copia los del directorio activo.

<nombre de archivo> es el nombre del archivo/s que se desea/n copiar. Admite caracteres comodín.

<unidad>, nombre de la unidad que contiene el disco backup. Siempre hay que especificar una unidad.

/A añade los archivos de backup al disco sin borrar los ya existentes.

/S copia todos los subdirectorios.

/M hace que sólo se copien aquellos archivos que han sufrido modificaciones desde la última sesión de copiado.

/D:<Fecha> copia los archivos que han sido modificados después de la fecha indicada.

BREAK

Tipo interno

Le permite indicar al DOS que compruebe si se ha efectuado una ruptura de control, siempre que un programa pida al DOS que realice alguna función.

Su sintaxis es:

BREAK [ON/OFF]

ON indicará que se compruebe si se está introduciendo un [Ctrl-Break] por el teclado.

CHDIR

Tipo interno

Cambia el directorio actual del DOS de la unidad especificada o asumida por omisión, o visualiza el camino del directorio actual por una unidad.

Su sintaxis es:

CHDIR <unidad> <ruta> o CD <unidad> <ruta>

para cambiar la ruta y:

CHDIR o CD

para visualizar la ruta actual.

CHKDSK

Tipo interno

Analiza los directorios, los archivos y la Tabla de Asignación de Archivos en la unidad designada o asumida por omisión, y produce un informe del estado de la memoria y del disco.

Su sintaxis es:

CHKDSK <unidad> <nombre archivo>/N/F

/V visualiza el nombre de cada uno de los directorios y archivos del disco.

/F indica al DOS que corrija los errores que detecte en el directorio, si así lo indica el usuario cuando aparezca ese error.

CLS

Tipo interno

Borra la pantalla. No lleva parámetros, por lo que su sintaxis es CLS.

COMP

Tipo externo

Compara el contenido del primer conjunto de archivos especificados con el contenido del segundo conjunto de archivos especificados. Admite caracteres comodín.

Su sintaxis es:

COMP <archivo/s 1> <archivo/s 2>

Su función es muy similar a la del comando DISKCOMP que veremos más adelante, sin embargo, COMP compara archivos entre dos discos, mientras que DISKCOMP compara los discos completos.

COPY

Tipo interno

Copia las salidas de un dispositivo o archivo en un archivo u otro dispositivo.

Ha sido ampliamente explicado, por lo que nos limitamos a resumir su sintaxis en:

COPY <origen> <destino>
COPY <archivo 1> <archivo 2>
COPY <archivo> <dispositivo>

CTTY

Tipo interno

Cambia la consola de entrada y salida estándar a una consola auxiliar o restaura el teclado y la pantalla como dispositivo de entrada/salida estándar.

Su sintaxis es:

CTTY <nombre dispositivo>

DATE

Tipo interno

Permite introducir la fecha o cambiar la que ya tiene el sistema. La fecha se graba en la entrada del directorio para cada archivo que se crea o altera, por lo que es muy importante tenerla actualizada.

Su sintaxis presenta dos opciones:

DATE

sin parámetros, el DOS indicará la fecha actual y preguntará la nueva fecha, si se pulsa [Intro] mantendrá la misma que tenía.

DATE <fecha>

Cambia directamente la fecha que tiene por la indicada. El formato de la fecha depende de la configuración que se le haya dado.

DEL

Tipo interno

Suprime el archivo o archivos especificado.

Su sintaxis permite caracteres comodín y es de la forma:

DEL <unidad:ruta archivo>

Si utiliza DEL *.* se suprimirán todos los archivos del disco, por lo que el DOS le preguntará antes de ejecutar el mandato:

Está seguro (S/N)?

DIR

Tipo interno

Hace un listado de todas las entradas de un directorio o sólo de aquellas que corresponden a archivos especificados.

Se puede especificar sólo un nombre de archivo, con lo que en pantalla aparecerá éste en caso de estar presente. Si el nombre lleva caracteres comodín, aparecerán los archivos especificados. Y si no se añaden parámetros, aparecerán todos los archivos existentes en el directorio del disco de la unidad especificada.

DISKCOMP

Tipo externo

Compara el contenido del diskette en la primera unidad con el camino al directorio que contiene el archivo del mandato.

Su sintaxis es:

DISKCOMP <unidad 1> <unidad 2> /1/8

Si no se pone unidad 2, se compara el contenido de unidad 1 con el de la unidad activa.

/1 especifica que se compare sólo la primera cara de los diskettes, aunque éstos sean de dos caras.

/8 comparará sólo 8 sectores por pista, aunque el diskette sea de 9 ó 15 sectores por pista.

Al terminar la comparación el DOS preguntará:

Comparar otro diskette (S/N)?

DISKCOPY

Tipo externo

Copia el contenido del diskette de la unidad origen en el diskette de la unidad destino. Si es necesario el diskette de la unidad destino adquiere el formato durante la copia. Esto puede resultar un ahorro de tiempo, porque evitamos el formateo, pero en la copia pueden transferirse todas las imperfecciones que posea el disco origen al destino.

Su sintaxis es:

DISKCOPY <origen> <destino> /1

/1 indica al DOS que sólo debe copiar la primera cara del diskette, independientemente del tipo de unidad o de diskette.

ERASE

Tipo interno

Borra un archivo específico o varios. Admite caracteres comodín.
Su sintaxis es:

ERASE <nombre de archivo>

Si no se especifica nombre de archivo se utiliza por omisión *.*, con lo que se borran todos los archivos del disco.

EXE2BIN

Tipo externo

Convierte archivos .EXE en archivos .COM o .BIN.
Su sintaxis es:

EXE2BIN <archivo origen> <archivo destino>

Si no se especifica lo contrario el DOS toma por omisión en el archivo origen la extensión .EXE y en el destino .BIN.

FDISK

Tipo externo

Es un programa que permite, a traves de los distintos menús que presenta, la creación de la partición del DOS y otras áreas con el disco fijo.

FIND

Tipo externo

Es un comando filtro, cuyo objetivo es las líneas del archivo <nombre de archivo> que contiene la secuencia de caracteres especificada en <cadena>. Si no se indica ningún archivo, la localización se realiza entre las líneas que se introduzcan mediante el dispositivo de entrada/salida estándar (el teclado). A no ser que se indique otra cosa, envía las líneas seleccionadas que contienen la cadena al monitor. Se puede extender la búsqueda a varios archivos, sin más que separar sus nombres mediante espacios en blanco.
Su sintaxis es:

FIND /V /C /N <cadena> <nombre de archivo>

/V selecciona las líneas que no contienen la cadena.

/C es un contador que indica tan sólo el número de líneas que contienen la cadena.

/N hace que al lado de cada línea seleccionada aparezca su número de introducción en el archivo.

FORMAT

Tipo externo

Inicializa el disco en la unidad designada o asumida por omisión con un formato de registro aceptable para el DOS; analiza el disco completo buscando las pistas defectuosas; y prepara el disco para la aceptación de los archivos del DOS mediante la creación del directorio, la Tabla de Asignación de Archivos y el cargador del sistema.

Su sintaxis es:

FORMAT <unidad> /V /1 /4 /8

Si no se especifica la unidad, el DOS toma la unidad activa por defecto, ¡CUIDADO!, en algunas versiones del DOS esto puede suponer formatear el disco del sistema con la consiguiente pérdida de todo su contenido.

/V indica que se le quiere dar al disco una etiqueta de volumen.

/1 formatea el diskette sólo por una cara, al margen del tipo de unidad.

/4 formatea un diskette de doble cara en uno de doble capacidad.

/8 para dar al diskette el formato correspondiente a 8 sectores por pista.

GRAFTABL

Tipo externo

Carga en memoria una tabla de datos de carácter adicional para el Adaptador de color gráficos. Posibilita al DOS para visualizar caracteres gráficos especiales, siempre y cuando el adaptador se encuentre en la posición gráficos.

GRAPFICS

Tipo externo

Permite imprimir el contenido de una pantalla del monitor de gráficos en una impresora cualquiera, dentro de una gama muy extensa, cuando se está utilizando un Adaptador de gráficos/color.

Su sintaxis es:

GRAPHICS <impresora> /R /B

Consulte en el manual de su impresora a cuál corresponde.

/R Para escribir en blanco y negro de forma positiva, si no se especifica se imprime un negativo con el blanco como negro y el negro como blanco.

/B para imprimir el color del contraste. Sólo es válido en algunos tipos de impresora.

KEYBXX

Tipo externo

Modifica el significado asignado a algunas de las teclas, según el idioma que se vaya a utilizar. Las dos últimas letras XX se sustituyen para identificar el país:

KEYBSP → España
KEYBUK → Reino Unido
KEYBGR → Alemania
KEYBFR → Francia
KEYBIT → Italia

LABEL

Tipo externo

Permite crear, cambiar y suprimir la etiqueta de volumen de un disco. Su sintaxis es:

LABEL <unidad> <etiqueta>

Si se omite la etiqueta el DOS preguntará:

Etiqueta de volumen (11 caracteres y/o Intro)

Si se pulsa Intro sin dar ningún nombre, la etiqueta existente queda suprimida.

MKDIR

Tipo externo

Crea un subdirectorio en un disco especificado.

Su sintaxis es:

MKDIR <unidad:ruta> o MD <unidad:ruta>

MODE

Tipo externo

Establece la modalidad de operación de una impresora o de un Adaptador de monitor de Color/Gráficos o del Adaptador de Comunicaciones asíncronas.

Su sintaxis admite varias posibilidades:

MODE <impresora> <anchura>,<espacio>

Establece la capacidad de cada línea y su separación al imprimirlas.

MODE <puerto> <baudios>, <paridad>, <bitsdatos>, <bitsparada>

Con este formato se controlan los parámetros de las comunicaciones de un puerto serie.

MODE <impresora>=<puerto>

Redirige hacia un puerto serie las salidas que normalmente se enviarían a las impresoras LPT1:, LPT2: O LPT3:.

MODE <método>

Elige el monitor activo, y determina el número de caracteres por línea y los colores que se van a utilizar en un monitor al que se ha conectado el adaptador de color/gráficos.

MODE <método>,R o L,T

Con esta configuración del comando MODE se puede centrar la imagen de un monitor que no sea IBM y que está conectado a un adaptador de color/gráficos.

MORE

Tipo externo

Es un comando filtro que lee los datos del dispositivo de entrada estándar, envía una pantalla completa de datos al dispositivo de salida estándar y luego hace una pausa con el mensaje —Más—.

Pulsando cualquier tecla de carácter se hace que se escriba otra pantalla completa de datos.

Se suele utilizar para revisar archivos muy extensos, o las salidas de un comando por la pantalla.

El siguiente ejemplo conecta la salida del mandato DIR con el filtro MORE, cuando se produzca una salida por pantalla completa aparecerá el mensaje —Más—. Si se desea continuar se pulsa una tecla.

A>DIR ˙MORE

PATH

Tipo interno

Hace que se busquen en los directorios especificados los mandatos o archivos por lotes que no se encontraron en el directorio actual. Puede especificarse una lista de unidades y de nombres de camino, separados por puntos y comas. Si no se incluye ninguna ruta, pero sí un punto y coma, el comando anulará todos los caminos. Si se omiten tanto la ruta como el punto y coma, es decir, sólo el comando PATH, se visualizarán todas las rutas definidas por el usuario.

Su sintaxis es:

PATH <unidad> <ruta>

PRINT

Tipo externo

Imprime una lista de archivos de datos en la impresora mientras se están realizando otras tareas con el sistema. Permite un máximo de 32 archivos para imprimir.

Su sintaxis es:

PRINT <nombre de archivo> /P /C /T /D:<impresora> /Q:<tamaño>

Si se desea más de un archivo, se separan éstos, para formar la lista o cola, por espacios en blanco.

/P establece la modalidad de impresión.

/C permite seleccionar la modalidad de cancelación, con lo que todos los archivos especificados se retirarán de la cola de espera para impresión. Si en el momento de dar el comando, el archivo indicado está siendo impreso, se detiene la impresión pasando el papel a la página siguiente.

/T establece la modalidad de terminación, cancelando toda la cola de impresión.

/D:<impresora> indica el dispositivo de impresión. Si no se especifi-

ca ninguno, el sistema asume el dispositivo por defecto PRN. Sólo es necesario especificarlo la primera vez que se utiliza el comando PRINT.

/Q:<tamaño> indica la cantidad de archivos que pueden existir en la cola con un límite entre 1 y 32. El valor por omisión es 10. Sólo es necesario especificar el parámetro la primera vez que se utiliza el comando PRINT.

PROMPT

Tipo externo

Establece un nuevo prompt del sistema.

Su sintaxis es:

PROMPT <cadena de caracteres>

La cadena de caracteres la constituyen agrupaciones de la forma $X, donde la X puede ser:

t saldrá la hora.
d saldrá la fecha.
p saldrá el directorio actual de la unidad activa.
v saldrá el número de versión.
n saldrá la unidad activa.
g saldrá el carácter >.
l saldrá el caracter <.
b saldrá el carácter ¨.
q saldrá el carácter =.
h realiza un retroceso y borrado del carácter anterior.
e saldrá el carácter ESCape.
_ saldrá la secuencia CR LF.

Si se omite la cadena el DOS toma por defecto ng que equivale al prompt conocido por todos del sistema.

RECOVER

Tipo externo

Recupera archivos procedentes de un disco que tenga un sector defectuoso. Puede recuperarse el archivo que contenga el sector no válido (menos los datos de dicho sector), o pueden recuperarse todos los archivos del disco si se dañó el directorio.

Su sintaxis es:

RECOVER <unidad:ruta archivo>

Si se introducen caracteres comodín sólo se recuperará el primero de los archivos encontrado.

RENAME

Tipo interno

Cambia el nombre del archivo especificado en el primer parámetro por el nombre y extensión dados en el segundo parámetro.

Su sintaxis es:

RENAME <nombre antiguo> <nombre nuevo>

o

REN <nombre antiguo> <nombre nuevo>

Nombre antiguo debe ser el nombre de un archivo existente en la unidad especificada.

Pueden utilizarse caracteres comodín para renombrar una serie de archivos.

RESTORE

Tipo externo

Devuelve archivos que fueron copiados de un disco a otro mediante el comando BACKUP.

Su sintaxis es:

RESTORE <unidad> <ruta> <nombre de archivo> /S /P

/S recupera, además, los archivos de los subdirectorios.

/P pide conformidad sobre los archivos recuperados y que hayan sufrido alguna modificación desde su copiado en el disco backup.

RMDIR

Tipo interno

Elimina un subdirectorio del disco especificado.

Su sintaxis es:

RMDIR <unidad:ruta> o RD <unida:ruta>

La única condición que debe cumplirse es que el directorio esté vacío

(con excepción de las entradas . y ..), no contenga archivos ocultos y no sea el directorio raíz o el actual.

SELECT

Tipo externo

Permite especificar el tipo de teclado y el formato de la fecha y la hora. Este comando utiliza el mandato DISKCOPY para copiar el diskette del DOS. Crea, además, dos archivos en el diskette copia del DOS:

— El archivo CONFIG.SYS que contiene el mandato COUNTRY=.
— El archivo AUTOEXEC.BAT que contiene el comando KEYBxx.

Su sintaxis es:

SELECT <código> <teclado>

El código y el teclado podrán elegirse entre los siguientes:

País	Código	Teclado
ESTADOS UNIDOS	001	US
FRANCIA	033	FR
ESPAÑA	034	SP
ITALIA	039	IT
REINO UNIDO	044	UK
ALEMANIA	049	GR

SET

Tipo interno

Este mandato inserta series de caracteres en el entorno del procesador de comandos (COMMAND.COM). El grupo completo de series que haya en el entorno queda disponible para todos los mandatos y aplicaciones.
Su sintaxis es:

SET <nombre=[parámetro]>

Si se introduce sólo SET se visualizará el conjunto actual de series del entorno.

SHARE

Tipo externo

Carga el soporte para compartir archivos.

SORT

Tipo externo

Es un comando filtro que lee líneas del dispositivo estándar de entrada, las dispone en el orden requerido, y las envía, ya ordenadas, al dispositivo estándar de salida. Si se escribe sin parámetros la ordenación se realiza por orden alfabético, o de menor a mayor número.

Su sintaxis es;

SORT/R/+<columna>

/R dispone las líneas en orden inverso al indicado.

/+<columna> establece la ordenación según la columna señalada. Este parámetro se utiliza para ordenar las líneas según el contenido de las columnas distintas de la primera.

SYS

Tipo externo

Transfiere los archivos del DOS IBMDOS.COM e IBMBIO.COM desde la unidad especificada como origen a la especificada como destino.

Su sintaxis es:

<unidad origen:ruta> SYS <unidad destino>

La unidad especificada como destino debe hallarse totalmente vacía, o haber sido formateada con FORMAT /S o /B. Esto se debe a que los dos archivos que copia el comando deben ocupar las dos primeras entradas del directorio, y porque IBMBIO.COM debe residir en sectores consecutivos del disco (se vio en el capítulo 2).

TIME

Tipo interno

Su función es similar a la de DATE, pero se refiere a la hora del sistema. Permite introducir o cambiar la hora conocida por el sistema. Siem-

pre que se crea o actualiza un archivo, se registra la hora en el directorio. Se puede cambiar la hora desde la consola o desde un archivo batch (.BAT).

Su sintaxis es:

TIME <hora:minutos> <:segundos> <.centésimas>

Si se omiten los parámetros el DOS nos mostrará la hora actual del sistema añadiendo el mensaje:

Introduzca nueva hora:

Si se pulsa [Intro] la hora no varía.

TREE

Tipo externo

Visualiza todos los caminos de directorio hallados en la unidad especificada y opcionalmente hace un listado de los archivos de cada subdirectorio.

Su sintaxis es:

TREE <unidad> /F

/F es opcional, se utiliza para visualizar los nombres de los archivos.

TYPE

Tipo interno

Muestra el contenido del archivo especificado en el dispositivo estándar de salida.

Su sintaxis es:

TYPE <nombre de archivo>

No permite la utilización de caracteres comodín.

VER

Tipo interno

No lleva parámetros. Su función es visualizar en el dispositivo estándar de salida el número de la versión del DOS con la que se está trabajando.

VERIFY

Tipo interno

Verifica que los datos que han sido grabados en un disco han sido correctamente registrados.

Su sintaxis es:

VERIFY <ON/OFF>

VOL

Tipo interno

Visualiza la identificación de volumen del disco de la unidad especificada.

Su sintaxis es:

VOL <unidad>

Si se omite la unidad, el DOS supone que se trata de la unidad activa.

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

INDICE GENERAL

5 AÑADA PERIFERICOS A SU ORDENADOR

Breve descripción de varios periféricos que facilitan la comunicación con el ordenador personal, con algunos ejemplos de fácil construcción: ratón, lápiz óptico, marco para pantalla táctil...

6 GRAFICOS ANIMADOS CON EL ORDENADOR

En este libro las técnicas utilizadas para la animación son el resultado de unas pocas ideas básicas muy sencillas de comporender. Descubrirá los trucos y secretos de movimientos, choques, rebotes, explosiones, disparos, saltos, etc.

7 JUEGOS INTELIGENTES EN MICROORDENADORES

Los ordenadores pueden enfrentarse de forma «inteligente» ante puzzles y otros tipos de juegos. Esto es posible gracias al nuevo enfoque que ha dado la IA a la tradicional teoría de juegos.

8 PERIFERICOS INTERACTIVOS PARA SU ORDENADOR

Descripción detallada de la forma de construir, paso a paso y en su propia casa, dispositivos electrónicos que aumentarán la potencia y facilidad de uso de su ordenador: tableta digitalizadora, convertidores de señales analógicas, comunicaciones entre ordenadores.

9 COMO HACER DIBUJOS TRIDIMENSIONALES EN EL ORDENADOR PERSONAL

Compruebe que también con su ordenador personal puede llegar a diseñar y calcular imágenes en tres dimensiones con técnicas semejantes a las utilizadas por los profesionales del dibujo con equipos mucho más sofisticados.

10 PRACTIQUE MATEMATICAS Y ESTADISTICA CON EL ORDENADOR

En este libro se repasan los principales conceptos de las Matemáticas y la Estadística, desde un punto de vista eminentemente práctico y para su aplicación al ordenador personal. Se basan los diferentes textos en la presentación de pequeños programas (que usted podrá introducir en su ordenador personal).

11 CRIPTOGRAFIA: LA OCULTACION DE MENSAJES Y EL ORDENADOR

En este libro se presentan las técnicas de mensajes a través de la criptografía desde los primeros tiempos hasta la actualidad, en que el uso de los computadores ha proporcionado la herramienta necesaria para llegar al desarrollo de esta técnica.

12 APL: LENGUAJE PARA PROGRAMADORES DIFERENTES

APL es un lenguaje muy potente que proporciona gran simplicidad en el desarrollo de programas y al mismo tiempo permite programar sin necesidad de conocer todos los elementos del lenguaje. Por ello es ideal para quienes reúnan imaginación y escasa formación en Informática.

13 ECONOMIA DOMESTICA CON EL ORDENADOR PERSONAL

Breve introducción a la contabilidad de doble partida y su aplicación al hogar, con explicaciones de cómo utilizar el ordenador personal para facilitar los cálculos, mediante un programa especialmente diseñado para ello.

14 COMO SIMULAR CIRCUITOS ELECTRONICOS EN EL ORDENADOR

Introducción a los diferentes métodos que se pueden emplear para simular y analizar circuitos electrónicos, mediante la utilización de diferentes lenguajes.

15 COMO CONSTRUIR SU PROPIO ORDENADOR

Cuando se trabaja con un ordenador, lo único que puede apreciarse, a simple vista, es una especie de caja negra que, misteriosamente, acepta una serie de instrucciones. En realidad, un ordenador es una máquina capaz de recibir, transformar, almacenar y suministrar datos.

16 EL ORDENADOR COMO INSTRUMENTO MUSICAL Y DE COMPOSICION

Análisis de cómo se puede utilizar el ordenador para la composición o interpretación de música. Libro eminentemente práctico, con numerosos ejemplos (que usted podrá practicar en su ordenador casero) y lleno de sugerencias para disfrutar haciendo de su ordenador un verdadero instrumento musical.

17 SISTEMAS OPERATIVOS: EL SISTEMA NERVIOSO DEL ORDENADOR

Características de diversos sistemas operativos utilizados en los ordenadores personales y caseros. Se trata de llegar al conocimiento, ameno aunque riguroso, de la misión del sistema operativo de su ordenador, para que usted consiga sacar mayor rendimiento a su equipo.

18 UNIX, EL ESTANDAR DE LOS SISTEMAS OPERATIVOS MULTIUSUARIO

La aparición y posterior difusión del sistema operativo UNIX supuso una revolución en el mercado, de tal modo que se ha convertido en el estándar de los sistemas multiusuario. Su aparente complejidad podría provocar, en principio, un primer rechazo, pero debido a su potencia se convierte rápidamente en una extraordinaria herramienta de trabajo apta para cualquier tipo de aplicaciones.

19 EL ORDENADOR Y LA ASTRONOMIA

Los cálculos astronómicos y el conocimiento del firmamento en un libro apasionante y curioso.

20 VISION ARTIFICIAL. TRATAMIENTO DE IMAGENES POR ORDENADOR

El procesado de imágenes es un campo de reciente y rápido desarrollo con importantes aplicaciones en área tan diversas como la mejora de imágenes biomédicas, robóticas, teledetección y otras aplicaciones industriales y militares. Se presentan los principios básicos, los sistemas y las técnicas de procesado más usuales.

21 PRACTIQUE HISTORIA Y GEOGRAFIA CON SU ORDENADOR

Libro interesante para los aficionados a estas ciencias, a quienes presenta una nueva visión de cómo utilizar el microordenador en su estudio.

22 LA CREATIVIDAD EN EL ORDENADOR. EXPERIENCIAS EN LOGO

El LOGO es un lenguaje enormemente capacitado para la creación principalmente gráfica y en especial para los niños. En este sentido se han desarrollado numerosas experiencias. En el libro se analizan estas experiencias y las posibilidades del LOGO en este sentido, así como su aplicación a su ordenador casero para que usted mismo (o con sus hijos) pueda repetirlas.

23 EL LENGUAJE C, PROXIMO A LA MAQUINA

Lenguaje de programación que se está imponiendo en los microordenadores más grandes, tanto por su facilidad de aprendizaje y uso, como por su enorme potencia y su adecuación a la programación estructurada. Vinculado íntimamente al sistema operativo UNIX es uno de los lenguajes de más futuro entre los que se utilizan los micros personales.

24 BASIC

El lenguaje BASIC es la forma más fácil de aprender las instrucciones más elementales con las que podemos mandar a nuestro ordenador que haga las más diversas tareas.

25 COMO ELEGIR UNA HOJA ELECTRONICA DE CALCULO

En este título se estudian las diferentes versiones existentes de esta aplicación típica, desde el punto de vista de su utilidad para, en función de las necesidades de cada usuario y del ordenador de que dispone, poder elegir aquella que más se adecúe a cada paso.

26 BASIC AVANZADO

Una vez conocidas las instrucciones fundamentales del lenguaje BASIC se plantea la cuestión de la realización de programas que resuelvan problemas o aplicaciones que se nos presentan diariamente en el trabajo, en casa o en los estudios. Este libro trata de mostrar cómo se podrían realizar algunas de estas aplicaciones, estudiando diversas estructuras que proporciona el lenguaje BASIC (como las subrutinas) y viendo las ideas fundamentales para realizar gráficos en pantalla mediante un programa y para almacenar datos en discos o cintas mediante los ficheros.

27 APLIQUE SU ORDENADOR A LAS CIENCIAS NATURALES

Ejemplos sencillos para practicar con el ordenador. Casos curiosos de la Naturaleza en forma de programas para su ordenador personal.

28 PRACTIQUE FISICA CON SU ORDENADOR

Deja que el ordenador te ayude en tus estudios. Materias tan difíciles como la Física, se ponen a tu alcance de una manera entretenida y mucho más clara, con programas que te permitirán entender las cosas desde un punto de vista más real.
Definiciones, fórmulas, gráficos y ejemplos, en un pequeño manual que puedas utilizar en cualquier momento.

29 PRACTIQUE QUIMICA CON SU ORDENADOR

En nuestra búsqueda particular de la «piedra filosofal», al modo de los antiguos alquimistas, nos ayudaremos del ordenador para que nos resulte más fácil. Con este libro conseguiremos entender fácilmente las valencias de los elementos, las reacciones Redox y las distintas teorías sobre el átomo. Nos servirá de guía para aprender la tabla periódica de los elementos y nos ayudará a comprender, mediante gráficos, una reacción en cadena. Podremos así convertir nuestra casa y nuestro ordenador en un gran laboratorio.

30 APRENDA MATEMATICAS Y ESTADISTICA CON EL LENGUAJE APL

APL es un lenguaje muy potente que proporciona gran simplicidad en el desarrollo de programas. Indudablemente, es mucho más apto que BASIC para la construcción de pequeños programas que realicen operaciones matemáticas de dificultad media, que además se expresan de una forma muy semejante a la notación matemática ordinaria, lo que lo hace fácilmente comprensible.

31 LOS LENGUAJES DE LA INTELIGENCIA ARTIFICIAL

Libro en que se describen los lenguajes específicos para la «elaboración del saber» y los entornos de programación correspondientes. El conocimiento de estos lenguajes, además de interesante en sí mismo, es sumamente útil para entender todo lo que la Inteligencia Artificial supondrá para el futuro de la Informática.

32 LA ESTACION TERMINAL PERSONAL

Las modernas técnicas de comunicación van permitiendo que las grandes capacidades de proceso y el acceso a bases de datos de gran tamaño estén cada día más al alcance de cada usuario (fuera ya de los centros de proceso de datos).

33 COBOL

Este libro pretende introducir al lector en uno de los lenguajes más utilizados y menos considerados del mundo informático. El Cobol es el lenguaje de gestión por excelencia y está presente en el desarrollo del software en la gran mayoría de empresas e instituciones públicas.

34 ADA

El considerable esfuerzo desarrollado por el Departamento de Defensa de los Estados Unidos (DoD) para que el lenguaje Ada fuese desarrollado quedará compensado por las aportaciones de este lenguaje a los sistemas informáticos del futuro.
Sus aplicaciones originales, sistemas en tiempo real para mando y control en el área de Defensa, han sido ampliadas al campo industrial para el control de procesos, aplicaciones en tiempo real, inteligencia artificial, etcétera.

35 EL ORDENADOR COMO MAQUINA DE ESCRIBIR INTELIGENTE

Descripción de algunos de los programas para tratamiento de textos existentes en el mercado, análisis comparativos y estudio de las posibilidades de cada uno de ellos. Guía práctica para la elección del procesador de textos que más se adecúe a nuestras necesidades y al ordenador personal del que dispongamos.

36 MS-DOS

El sistema operativo de muchos ordenadores personales es el sistema operativo de disco de Microsoft, más conocido como MS-DOS, que recibe su nombre de su principal actividad: manejar los discos y archivos de discos. Su conocimiento puede llegar a ser tan profundo como deseemos, las nociones básicas, sin embargo, pueden llegar a ser imprescindibles para el manejo de nuestro ordenador.

37 REDES DE AREA LOCAL

El objetivo de este libro es el de proporcionar al lector un conocimiento claro de lo que son las redes locales, de su tecnología, problemática y futuro, de forma que, si lo desea, pueda profundizar posteriormente, por medio de bibliografía especializada o por la práctica profesional.

38 LOS FUNDAMENTOS DE LA GRAFOLOGIA APLICADA Y SU POSIBLE TRATAMIENTO CON UN ORDENADOR PERSONAL

Se presentan en este libro los perfiles grafológicos óptimos correspondientes a diversas actividades laborales, así como los programas de ordenador necesarios para el manejo de estos datos. Obra eminentemente práctica y de aplicación de los conceptos teóricos desarrollados en ella.

39 EL ORDENADOR Y LA LITERATURA

En este libro se examinan procesadores de textos, programas de análisis literario y una curiosa aplicación desarrollada por el autor: APOLO, un programa que compone estructuras poéticas.

40 ¿MAQUINAS MAS EXPERTAS QUE LOS HOMBRES?

Después de situar los «sistemas expertos» en el contexto de la Inteligencia Artificial y describir su construcción, su funcionamiento, su utilidad, etc., se analiza el papel que pueden tener en el futuro (y en el presente, ya) de la Informática, así como los polémicos temas de la «capacidad para desbancar a la inteligencia humana», y las posibilidades de «aprender» de que se puede dotar a un procesador, etcétera.

NOTA:
Ediciones Siglo Cultural, S. A., se reserva el derecho de modificar, sin previo aviso, el orden, título o contenido de cualquier volumen de esta colección.

Acercarse al mundo que rodea al DOS supone un adelanto en el conocimiento de los ordenadores personales. En la actualidad, este sistema operativo se ha convertido en el más importante del mercado, por sus grandes posibilidades.

Conocer el DOS no quiere decir que tengamos que ser expertos en él, sin embargo, sí es aconsejable conocer sus posibilidades y aprender a manejarlo. En este libro encontrará esa base que le permitirá convertirse en un nuevo usuario del DOS.